Algorithms for Protein Comparative Modelling and Some Evolutionary Implications

BRUNO CONTRERAS-MOREIRA

2003

Biomolecular Modelling Laboratory
Cancer Research UK, London Research Institute
44 Lincoln's Inn Fields, London, WC2A 3PX

and

Department of Biochemistry and Molecular Biology
University College London
Gower Street, London, WC2E 6BT

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Biochemistry of the University of London.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

A mi familia, incluida Pili

Abstract

Protein comparative modelling (CM) is a predictive technique to build an atomic model for a polypeptide chain, based on the experimentally determined structures of related proteins (templates). It is widely used in Structural Biology, with applications ranging from mutation analysis, protein and drug design to function prediction and analysis, particularly when there are no experimental structures of the protein of interest. Therefore, CM is an important tool to process the amount of data generated by genomic projects. Several problems affect the performance of CM and therefore solutions for them are needed to increase its applicability. In this work different algorithms and approaches were tested with this aim, particularly to help in template selection and alignment, and some useful insights were obtained.

First, this work describes the development of DomainFishing, a tool to split protein sequences into functionally and structurally defined domains and to align each of them to the available templates. The performance of our approach is benchmarked and some problems and possible developments are identified. When comparing different alignment procedures none of them is found to be consistently superior, suggesting that a combination of several could be an advantage. Driven by these ideas and the fact that selecting templates can be a difficult problem, a new modelling approach is designed and tested. This algorithm uses crossover, mutation and selection within populations of protein models generated from different templates and alignments to obtain recombinant structures optimised in terms of fitness. Despite our simple definition of fitness, the procedure is shown to be robust to some alignment errors while simplifying the task of selecting templates, making it a good candidate for automatic building of reliable protein models. In-house benchmarks of the method show its strengths and limitations. The method was also benchmarked during the fifth Critical Assessment of techniques for protein Structure Prediction (CASP5), in which its perfomance was encouraging both for comparative modelling and fold recognition targets, among the top 20 predictors. Finally, we present some data to support a possible evolutionary feedback mechanism between protein structure and gene structure, using human and murine genomic data, structural data from the Protein Data Bank and the protein recombination methodology. This may have some implications for understanding protein evolution and protein design, which are discussed.

Acknowledgements

Replying by e-mail to an advertisement that I found in a magazine almost four years ago had unexpected consequences: I ended up in London. Just like in *Rayuela* (a book by Julio Cortázar) now I must acknowledge people from both this and the other side of the British Channel.

From this side (el lado de acá), I am specially grateful to Paul, my supervisor, for all our discussions and arguments. Thanks for your patience and guidance. From the Biomolecular Modelling Lab (BMM) I must also thank my friend Paul Fitzjohn for his generous help and for teaching me so many things. Graham also helped me a lot, particularly with Maths, Molecular Dynamics, English and even Spanish. Thanks for all that and for your support. I should also acknowledge the latest signings, Chris, Pall and José for our joint projects, their advice and for sharing their *readPDB* routines. From the old Mike Sternberg's BMM, I am indebted to Arne for many things, including introducing me to LaTeX and Bagpuss and also for TeXMed. I am sure I still owe you and Aengus a couple of pints. This is probably true for Adrian, Lawrence and Ben as well. I should also put here Raphael and Dave for everything, including reading drafts of this thesis and for being my loyal friends outside the lab (pubs are among the best places to talk about anything).

Of course I should thank my overseas family (María, Afroditula, Joana, Nick, Themis and Óscar) just for being there every day and for the Spanish, Greek, Portuguese and Lancashire food. Sofiki should also be here, but she preferred to live at Baron's Court.

I would also like to thank Cancer Research UK (and its volunteers) for the funding, the good working environment and the great time I had here. Neil McDonald and Nancy Hogg were especially kind to me. Thanks for your time and support.

From the other side (el lado de allá), I must remember first my family. Living with them has so far been the best school. Álex, Edu, Elena, Javivi, Jevi and Arancha have always been my friends despite the distance. Agus put me on this crazy track and still inspires me. On a more scientific side of things, I thank here Antonio García-Bellido and Alfonso Valencia (and their groups, including Cassandrex, Damien and 'cubanito') for helping me reaching this page. My colleagues from www.precarios.org deserve my recognition for all their work to make life easier for Spanish scientists.

Finally, Pili deserves everything for being at both sides and never surrendering.

List of abbreviations

BLAST	Basic Local Alignment Search Tool
BLOCKS	'alignment Blocks' (no abbreviation)
BLOSUM	Blocks Substitution Matrix
CASP	Critical Assessment of techniques for protein Structure Prediction
CM	Comparative Modelling
DNA	Deoxyribonucleic Acid
FASTA	Fast Alignment Search Tool
FR	Fold Recognition
HMM	Hidden Markov Model
HSP	High-scoring Segment Pair
HTML	Hypertext Markup Language
IEB	Intron-Exon Boundary
IMPALA	Integrating Matrix Profiles And Local Alignments
NCBI	National Center for Biotechnology Information
NMR	Nuclear Magnetic Resonnance
nr	Non-Redundant Protein Database compiled by the NCBI
mRNA	messenger Ribonucleic Acid
ORF	Open Reading Frame
PAM	Point Accepted Mutation
PDB	Protein Data Bank
PFAM	Protein Family database of alignments and HMMs
PIR	Protein Information Resource
ProDom	Protein Domain
PROSITE	not an abbreviation (protein sequence pattern database)
PSI-BLAST	Position Specific Iterated BLAST
PSI-PRED	Position Specific Iterated Prediction
PSSM	Position Specific Scoring Matrix
RMSD	Root Mean Square Deviation
RNA	Ribonucleic Acid
SCOP	Structural Classification Of Proteins
URL	Unified Resource Locator

Table 1: List of abbreviations used throughout this document.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A monkey is a machine that preserves genes up trees, a fish is a machine that preserves genes in water; there is even a small worm that preserves genes in German beer mats. DNA works in mysterious ways.

RICHARD DAWKINS

What about proteins? According to the Central Dogma of Molecular Biology, genes are just portions of double-stranded molecules of deoxyribonucleic acid (DNA), but their information must be faithfully transcribed to single-stranded ribonucleic acid (RNA) molecules, and finally translated to proteins, to be used. Proteins are polymers of amino acids whose composition is encoded in genes. While genes have limited direct influence on cellular processes, proteins are responsible for the shape and structure of cells and serve as the main instruments for molecular recognition and catalysis of chemical reactions. Understanding proteins is therefore essential to understand cellular mechanisms, and in general, to understand life.

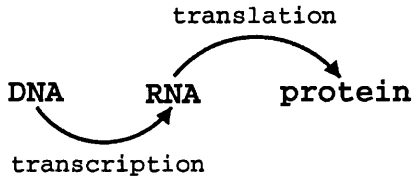


Figure 1.1: The Central Dogma of Molecular Biology. Portions of DNA sequence are copied into transient RNA molecules. This messenger RNA drives protein synthesis.

Proteins are encoded in genes using an universal code, the genetic code, deciphered in the early 1960s. Each gene precisely defines the amino acid sequence of a protein, allow-

ing the cell machinery to synthesise proteins following the genetic recipe. This synthesis consists of covalently bonding amino acids on a one by one basis to a growing polypeptide chain. Finally, the chain must adopt the right shape, called the native state, to be functional in the cellular context. This is the folding pathway, which builds compact protein domain(s) from a linear polypeptide chain by forming non-covalent interactions. When a protein unfolds, in a process called denaturation, its covalent backbone structure remains intact, the sequence of amino acids is still the same, but loses its biological activity. Thus, the three-dimensional structure of a protein determines its function.

In small proteins, as shown experimentally by (Anfinsen *et al.*, 1961), the denaturation reaction is reversible(Dobson & Karplus, 1999). For example, unfolded ribonuclease (an enzyme that cleaves RNA molecules) can fold again *in vitro* just by removing the denaturing agents. This simple experiment shows that the folding reaction for ribonuclease is autonomous: its fold is a consequence of its sequence. For many other proteins, things are more complex. For instance chaperonins (another class of proteins) may be required for the correct folding reaction (Hartl & Hayer-Hartl, 2002). In either case, the folding process takes between 0.1 and 1000 seconds (Branden & Tooze, 1999). This short time suggests that the folding process is not a blind exploration, since that would require in the order of 10^{50} years to complete for a medium size protein (this is known as the Levinthal paradox).

The folding process is important for our understanding of proteins and for the possible applications of proteins in technology. But despite considerable efforts over more than 40 years, the folding process remains an unsolved problem. There is no efficient algorithm to accurately fold a polypeptide *ab initio*, despite the fact that computers are, at least, doubling their speed every two years. Reasonably accurate protein models can be obtained, though, by using related experimental information together with comparative modelling algorithms. The main motivation of this thesis is to improve on these comparative methods to build molecular models of proteins, particularly to investigate the function and evolution of many proteins found in large-scale sequencing projects for which no experimental data is available. Due to the amount of data and the calculations usually required, these methods can only be implemented as computer programs.

1.1 Protein structure and function

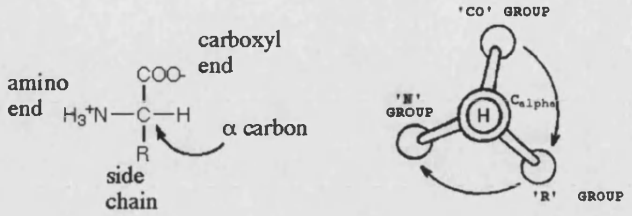
The structural features of folded proteins can be analysed in a hierarchy of complexity consisting of up to four layers. The primary structure is the simplest, represented by the amino acid sequence; the quaternary structure is the most complex, as spatial arrangements of different polypeptide chains occur at this level. Here they are briefly explained.

1.1.1 Primary structure

The primary structure of a protein is the linear sequence of amino acids as codified by the corresponding gene. To be more precise, an expressed gene is transcribed to compose a messenger RNA (mRNA). This molecule contains that gene's particular sequence of nucleotides, using an alphabet of four different nucleotides. The translation machinery searches for an open reading frame within the mRNA and starts protein synthesis by adding one amino acid every three RNA nucleotides, according to the genetic code. The mRNA also contains information to stop the synthesis. The genetic code is almost universal and the general mechanism is conserved in every organisms, although there are differences between prokaryotes and eukaryotes. In many cases, mostly in eukaryotes, the mRNA must be processed before translation since it contains *introns*, fragments that are not supposed to be translated. They must be removed to put *exons* together in a linear molecule (see Section 1.1.5).

There are 20 naturally occurring amino acids and they share a common composition (see Figure 1.2): an amino group and a carboxyl group joined by a single carbon, known as the α carbon, from which different side-chains are attached. In the case of Glycine, the side-chain is a single hydrogen atom.

A polypeptide chain contains n amino acids forming covalent peptide bonds between the carboxyl group of residue $i-1$ and the amino group of residue i, as shown in Figure 1.3. Due to its delocalised nature, this bond is rigid and planar. The bond immediately before the peptide bond can rotate, as well as the bond immediately after. The angles of rotation of these two bonds are called ϕ(phi) and ψ(psi). The backbone of a protein is the polypeptide chain after stripping the side-chains, and can be accurately described in terms of ϕ, ψ angles. The conformation of the backbone is dictated mainly by the different chemical properties of the side-chains and their interactions with the backbone. Side-chains can be classified as non-polar, polar and charged.



(a) (b)

Figure 1.2: The general structure of amino acids. (a) Description of groups bonded to the α carbon. Side-chain atoms are named with Greek letters following the α carbon. The first carbon of the side-chain is thus the β carbon, or simply $C\beta$. (b) Stereochemistry of amino acids, with the H atom pointing out the plane. Natural amino acids are L stereoisomers (that rotate plane polarized light to the left). Using the CORN rule, in L amino acids the path from CO to N passing through R is done clockwise. (Taken from http://web.mit.edu and http://www.friedli.com)

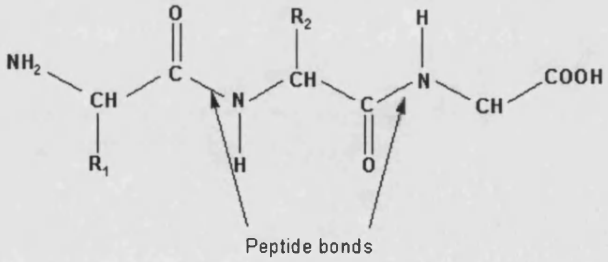


Figure 1.3: The peptide bond. (Taken from http://www.friedli.com)

1.1.2 Secondary structure

To neutralise the polar charges of the backbone, proteins adopt conformations that maximise local interactions through hydrogen bonds. There are two main types of secondary structure (SS) found in proteins: α-helices (right handed) and β-strands, as predicted by Ramachandran & Sasisekharan (1968) by studying the sterical limitations to the ϕ, ψ rotation. There are other less represented secondary structures, such as the β-turns (Hutchinson & Thornton, 1994), but more importantly, there are regions in proteins without a regular secondary structure. They are called generically *loops*, but there is no precise definition for them.

In α-helices, carboxyl and amino groups of residues i and $i+4$ form hydrogen bonds to complete one turn every 3.6 residues. Helices comprise continuous regions in the se-

quence, requiring at least 4 residues. β-sheets are built from a combination of several regions of the polypeptide chain, unlike helices, called β-strands. Strands are usually from 5 to 10 residues long and are in an almost fully extended ϕ, ψ conformation. Adjacent strands align forming hydrogen bonds between carboxyl and amino groups. The formation of secondary structure depends to a large extent on the primary structure, since some residues favour the formation of helices or strands and others favour loops. Indeed sequence information is enough to predict secondary structure with remarkable accuracy (check for example http://cubic.bioc.columbia.edu/eva (Eyrich *et al.*, 2001)) or even folding rates for simple proteins (Gong *et al.*, 2003).

1.1.3 Tertiary and quaternary structure

Most proteins form compact globules, usually consisting of secondary structure elements connected by loops (see Section 1.1.4 for non-globular proteins). This folding unit is called a *domain*. The interior of a domain contains mainly hydrophobic side-chains, whilst loops tend to be exposed to the solvent (Branden & Tooze, 1999). Nearly all protein structures solved so far show this trend and indeed it has been proposed that burying hydrophobic parts of proteins is the main driving force in folding (Dill, 1990). Apart from hydrophobic interactions, there are other interactions stabilising the tertiary structure (Lehninger, 1982):

- Hydrogen bonds between adjacent loops.
- Ionic interactions between oppositely charged side-chain groups (salt bridges).
- Disulfide bridges between cysteine residues close in space. Specific enzymes may assist in this task.

The native tertiary structure of a protein is the stablest form in solution but is not rigid. Many proteins exhibit flexibility and indeed their function may depended on conformational changes.

Domains are also the functional units of proteins, although a polypeptide chain may have several domains. Sometimes domains are only active in their biological context when they form multimeric complexes. These specific associations of identical domains occur at the quaternary structure level. For example, haemoglobin is a tetramer in which monomers work cooperatively (See Figure 1.4).

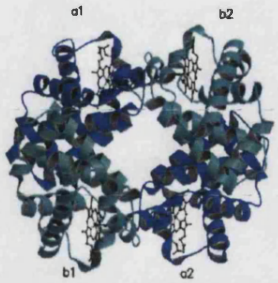


Figure 1.4: Haemoglobin tetramer. Secondary structure elements (helices) and globular domains (two green and two blue) can be identified. Associated heme groups are shown in black. (Taken from http://www-cryst.bioc.cam.ac.uk)

It is important to note that proteins with significant similarity at the sequence level have similar tertiary structures. However, structural similarity can often be mantained between evolutionarily related proteins despise the loss of significant similarity at the sequence level.

1.1.4 Fibrous and membrane proteins

Fibrous proteins are structural support materials, usually built up from long fibers. Instead of being made of compact domains, they form polymers by cross-linking or interleaving monomers. Examples are keratin, collagen and silks. Keratin is made of coiled-coil helices, collagen is a triple helix (made of proline-rich helices) and silk is a fiber of β-sheets. Organelles and cell membranes incorporate protein molecules, either spanning the membrane or anchoring to it. Because these proteins are functional on the membrane, and because their fold is stabilised by the lipid interactions, it is often difficult to determine their structure outside the membrane context. For this reason our structural knowledge about membrane proteins is relatively poor. However, we do know that membrane spanning proteins can be made of α-helices, in the case of bacteriorhodopsin, or β-sheets, in the case of porins. Membrane proteins still represent a technical challenge.

1.1.5 Evolution of proteins: introns and exons

In 1977 Phillip Sharp and Richard Roberts found that eukaryotic genes can be split up by non-coding DNA segments, that are removed after transcription (see Figure 1.5). Wal-

ter Gilbert coined the terms for these segments, *introns*, interrupting the coding *exons* (Gilbert, 1978). This discovery led to a search to see how prevalent they are. Introns are widespread in eukaryotes but they are quite rare in prokaryotes. This has prompted speculation about the evolution of organisms in general and the role introns may have in it.

Figure 1.5: Splicing of mRNA: removing of introns. Eukaryotic genes can be viewed as arrays of coding segments (exons, shown as E1,E2 & E3) that can be split by non-coding segments (introns, I1 & I2). Introns are usually removed during mRNA processing, before translation takes place, and therefore do not code for any part of the new protein molecule.

There are several types of introns, but basically some are auto-splicing and some are spliced by specific cell machinery. Either way, most prokaryotes lack them, with the remarkable exception of some archeobacteria, supposed to be really ancient forms of life. These facts allow two contradictory theories to coexist: the *introns-late* and the *introns-early* (Stoltzfus *et al.*, 1994).

The introns-early theory proposes that:

1. Exons are the descendants of ancient mini-genes and introns are the descendants of the spacers in between.
2. Contemporary proteins were first assembled from sets of exons.
3. The machinery of splicing originated in an ancient RNA world.
4. Introns were lost completely from bacteria as well as several protists groups.

In contrast, the introns-late theory states that:

1. Split genes arise from uninterrupted genes by insertion of introns.
2. Contemporary proteins probably first arose without the participation of introns.
3. The spliceosome machinery originated from fragmented self-splicing introns.

4. Spliceosomal introns never existed in the ancestors of today's organisms that lack them.

Conservation studies across species support the existence of early introns, conserved throughout evolution. On the contrary, non-conserved introns are more likely to habe been acquired more recently (Fedorov *et al.*, 2001, 2002). To enhance their models, evidence from protein structure analysis has also been used by supporters of both ideas (Stoltzfus *et al.*, 1994; de Souza. *et al.*, 1996, 1997), so that it is still not clear whether these theories are complementary or contradictory, though both seem to be possible, in the view of de Souza. *et al.* (1998).

1.2 Experimental methods for determination of protein structure

The two main techniques for experimental determination of protein structure are X-ray crystallography and nuclear magnetic resonance (NMR). Both methods collect atomic information with coordinate errors below 3Å (Rhodes, 2000). Complementary techniques can also be used, such as circular dichroism spectroscopy, to get information about the secondary structure content in a protein (Johnson, 1990). A different set of techniques are those related to electron microscopy and tomography, which can obtain images of large molecular complexes, membrane proteins or even virus capsids. The resolution of structures built with these developing techniques can be as good as 3.5Å (Henderson *et al.*, 1990), but falls usually in the range 8-20Å. Other techniques can be used to obtain valuable structural information, such as mass spectroscopy, fluorescence resonance energy transfer techniques, site-directed mutagenesis, yeast two-hybrid assays or protein arrays (Sali *et al.*, 2003).

1.2.1 X-ray crystallography

At present, this is the main experimental technique used in Structural Biology. Around 85% of all solved protein structures have been obtained using this technique. The technique basically consists of growing crystals of the protein of interest and subsequently using them to diffract a X-ray beam. The diffraction patterns are recorded and used to reconstruct the three-dimensional structure of the protein, by applying Bragg's law and Fourier transformations. The quality of the crystals directly affects the quality of the models derived from them, and sometimes flexible parts, such as exposed loops, cannot be re-

solved at all. The main advantage of this approach is the accuracy of the models obtained. The best X-ray models can have average atomic errors well under 1Å, making them ideal for rational drug design experiments. The main difficulty is the need to grow ordered protein crystals, making this step the bottleneck of the whole procedure. The same protein can be crystallised in different conditions and crystal lattices, yielding molecular models that deviate, on average, 0.6Å on their backbone coordinates (Montelione *et al.*, 2000).

There are currently international large-scale efforts to solve protein structures, the so called Structural Genomic projects (see 1.6).

1.2.2 NMR

This technique is based on the magnetic moments of some atomic nuclei such as ^{1}H, ^{13}C,^{15}N,^{31}P. If proteins containing these isotopes are analysed under a magnetic field, the chemical environment of atoms containing these nuclei can be probed and inter-atomic distances in the molecule derived. Sequential assignment methods developed by Wütrich and his group (see for example (Wagner & Wuthrich, 1982)) map distance constraints to the sequence and, finally, three-dimensional models based on them are built. Usually a set of different models can be obtained, all of them compatible with the experimental data. When comparing NMR models with models obtained by X-ray crystallography, they usually show backbone root mean square deviations (RMSD) around 1Å(Shaanan *et al.*, 1992).

The main limitations of NMR are obtaining highly concentrated protein solutions at acidic pH values and solving proteins longer than 300 residues, although recent achievements, such as the analysis of the GroEL complex (Fiaux *et al.*, 2002), suggest that large sizes are becoming less of a problem. The major advantage is that NMR is more suitable than X-ray crystallography to study dynamic processes in proteins.

1.3 Theoretical methods to model protein structure and dynamics

Before some of the methods are introduced, it is important to underline the importance of protein databases, on which many of them rely.

1.3.1 Databases

There is a wealth of knowledge that has been accumulated over the years about proteins. Sequence, enzymatic activity, phenotypes, mutations, evolutionary analysis are all data that can be used to understand a protein's structure and function. Many databases have been constructed to organise this data and allow easy access through the Internet. Here the data resources most extensively used in this work are presented. The URLs for these resources can be found in Appendix B.

Protein Data Bank (PDB)

The Protein Data Bank is a worldwide repository for the processing and distribution of three-dimensional biological macromolecular structure data (Berman *et al.*, 2000). As of April 2003 it contained 20473 structures obtained mainly by X-ray or NMR technologies. The PDB also contains theoretical molecular models.

Structural Classification of Proteins (SCOP)

The SCOP database, created by manual inspection and automated methods, is a hierarchical database that aims to provide a detailed and comprehensive description of the structural and evolutionary relationships between all proteins whose structure is known (Murzin *et al.*, 1995). As such, it is a valuable resource to accompany the PDB. The CATH Protein structure classification (Orengo *et al.*, 1997) is a similar resource but has not been used in this work.
Sequences for SCOP domains can be obtained from ASTRAL (Brenner *et al.*, 2000), as well as non-redundant subsets.

Protein families database (PFAM)

Pfam is a large collection of multiple sequence alignments covering many common protein domains and families. Each Pfam family contains a multiple alignment, domain architecture, information about species distribution, links to other databases and known protein structures from the PDB (Bateman *et al.*, 2002). Families are generated by extending a hidden Markov model (Baldi *et al.*, 1994; Eddy, 1996) of a manually aligned group of amino acid sequences, the *seed* (see also Section 2.1.3). This set is called Pfam-A and contains 5193 families in its 8.0 release (February,2003). To further increase the coverage of sequences, the Pfam team provide an automatically generated supplement called Pfam-B, containing a large number of small families taken from the ProDom database (Corpet

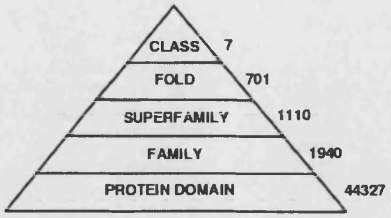


Figure 1.6: The SCOP classification, version 1.61 (September,2002). The CLASS level at the top of the triangle is the most general classification level. The first four classes are most relevant for this work, as they include α, β, α/β and $\alpha+\beta$ folds. Several entries from a level can be summarised by the next higher level (e.g. a FOLD contains one ore more SUPERFAMILIES). The lowest level is PROTEIN DOMAIN. The numbers of distinct entries at each level are given, making a total of 44327 domains (including the same domain in different species), extracted from 17406 PDB entries.

et al., 2000) that may partially overlap with Pfam-A.

Sequence databases

In general, the database used in this work for sequence similarity searches was *nr*, the weekly updated non-redundant protein sequence database generated at the U.S. National Center for Biotechnology Information (NCBI). This database is based on GenBank DNA coding sequences (Benson *et al.*, 2002), the PDB, SwissProt (Boeckmann *et al.*, 2003), PIR (Wu *et al.*, 2003) and the Japanese Protein Research Foundation database.
Whenever human genomic data was used, it was downloaded from Ensembl (Clamp *et al.*, 2003).

1.3.2 Introduction to algorithms

Algorithms are methods for solving problems which are especially suited for computer implementation (Sedgewick, 1988). Algorithms provide general solutions to general problems. Often several algorithms can be applied to solve the same problem and this certainly happens in the field of Structural Biology. In this section I will briefly introduce some algorithms concerning this work, but first it may be necessary to define a few important terms regarding the description of algorithms.

An algorithm is said to be *deterministic* if precisely the same steps are needed to solve a given initial problem, yielding exactly the same solution. On the contrary, *stochastic* algorithms have random components and therefore they will provide different answers,

though probably similar, to the same problem. Some algorithms have been proven to be correct by using mathematical induction. For the right sort of problems, these algorithms always work. However, frequently such algorithms cannot be found or they are too slow to compute and approximations are preferred. These less rigorous methods generally incorporate the current empirical knowledge about the problem of interest and are generically called *heuristic* methods.

Here a few families of algorithms are listed:

- Exhaustive searches that implement the search space as a tree and explore every branch, because no deterministic way is known to get quickly to the solution. However, there are pruning techniques to reduce the space that needs to be explored, improving the performance of these procedures. The size of the search tree is a reasonable estimate of the computing time required to find a solution. These approaches are very expensive in computing terms and are sometimes called brute force algorithms (Gonzalo-Arroyo & Rodríguez-Artacho, 1997).

- Monte Carlo techniques, which randomly sample by use of probability functions to perform statistical simulations. These are stochastic methods in which the accuracy of the estimates, by means of the density of sampling, can be controlled, affecting the required computing time (Press *et al.*, 1992).

- Dynamic programming, a family of algorithms that apply the recursive principle of divide-and-conquer to the extreme. Basically the problem is split into all the possible subproblems and all of them are solved and stored to compose global solutions, making the whole procedure significantly time consuming, but allowing efficient computation of different solutions (Sedgewick, 1988).

- Genetic algorithms, that imitate the principles of natural evolution to apply selection, according to some fitness estimate, within populations. They have been applied to a whole variety of optimisation problems (Michalewicz, 1996).

- Simulated annealing, a Thermodynamics-inspired algorithm to look for a global extremum in multi-dimension functions. These methods use the Boltzmann probability distribution during a hypothetical process of cooling down an initially hot system to select acceptable random jumps in space (Press *et al.*, 1992).

- Steepest descent and conjugate gradient methods, which use local gradients of the function to be optimised to guide each step in the path to possible global extrema (Leach, 2001).

1.3.3 Overview of algorithms in protein structure prediction

The large variety of problems and subproblems in this field has attracted scientists to consider a broad scope of algorithms to solve them. Three main approaches for protein structure prediction can be outlined:

- *Ab initio*, classically defined as the folding of the protein sequence according to physical principles. In recent years this approach has made use of techniques to assembly protein conformations from small unrelated peptides (see for instance (Simons *et al.*, 1997a) and (Jones, 2001)).
- *Fold recognition* (or threading), recognising that a protein sequence may represent a protein fold already classified by experimental techniques. Seminal contributions to this were Sippl (1990),Bowie *et al.* (1991) and Jones *et al.* (1992).
- *Comparative modelling*, in which proteins from the PDB, assumed to be homologous to the query, are used to guide the building of a three-dimensional atomic model. Greer (1981), Jones & Thirup (1986) and Sutcliffe *et al.* (1987a) pioneered this particular methodology.

1.3.4 Overview of protein minimisation and dynamics

Molecular dynamics studies of proteins are based on the molecular mechanics framework, which uses empirical force fields to calculate intra- and inter-molecular forces within a system. At this level of detail atoms are the elementary components of the system, allowing these calculations to be much faster than equivalent quantum mechanics calculations (Leach, 2001). Force fields contain empirically obtained reference values for four types of parameters: bond stretching, angle bending, torsional terms and non-bonded interactions. Force fields should in principle be transferable sets of parameters, extracted for example from a few proteins or from a collection of small molecules, to be then be applied to many different macromolecules. Examples of force fields with parameters for biological molecules are CHARMM (Brooks *et al.*, 1983), AMBER (Cornell *et al.*, 1995) and OPLS (Damm *et al.*, 1997). Energy estimates of molecular systems can be obtained with functions that include some of these generic terms:

$$P(r^N) = w_1 \cdot bond(r^N) + w_2 \cdot angle(r^N) + w_3 \cdot torsion(r^N) + w_4 \cdot non\_bonded(r^N) \quad (1.1)$$

where w_1, w_2, w_3, w_4 are weights for each term and $P(r^N)$ is the potential energy of N atoms in positions r_n.

By minimising $P(r^N)$ functions one can search for the global steric minimum of a given molecule, and that would be in theory the most likely conformation, the most stable, for it. Unfortunately, these multidimensional potential functions have many local minima and, in addition, force fields are not always transferable. As a consequence, minimised conformations do not always reproduce conformations observed experimentally.

The derivative of the potential of one atom in a molecule with respect to its position r_n is the force that it is receiving from the rest of the molecule. This is the principle for the simulation of molecules by Molecular Dynamics (MD), in which the behaviour of a molecular system is monitored along a given period of time through small (fs) time steps (Leach, 2001).

1.4 Comparative modelling of proteins (CM)

A more detailed introduction to this topic is now given, since it was the inspiration for most of the work in this thesis. Paul W.Fitzjohn is acknowledged here for his help, particularly with Section 1.4.5.

A generic flowchart for CM methods used by most developers in the field can be seen in Figure 1.7. The steps shown are common to the two main modelling protocols: satisfaction of spatial restraints (Sali & Blundell, 1993) and building up a protein by inheriting segments of other proteins (Greer, 1981; Jones & Thirup, 1986; Sutcliffe *et al.*, 1987a). However, some steps may be executed concurrently or in a different order.

1.4.1 Finding the best templates

Templates can be found by sequence similarity alone, or by using additional sources of structural information, such as secondary structure. The former approach is used by the BLAST (Altschul *et al.*, 1997) and FASTA (Pearson & Lipman, 1988) families of programs, where a query sequence is scanned against a database of template sequences using broad-spectrum matrices, such as BLOSUM (Henikoff & Henikoff, 1993) or PAM (Schwartz & Dayhoff, 1978), to score the alignments. Increased sensitivity can be gained by using the information of protein families (represented as position specific scoring matrices or hidden Markov models) as family-specific matrices, and by using intermediate sequences searching procedures(Baldi *et al.*, 1994; Krogh *et al.*, 1994; Eddy, 1996; Park *et al.*, 1998; Schaffer *et al.*, 2001). Further sensitivity can sometimes be gained by including structural information such as residue solvent accessibility and secondary structure

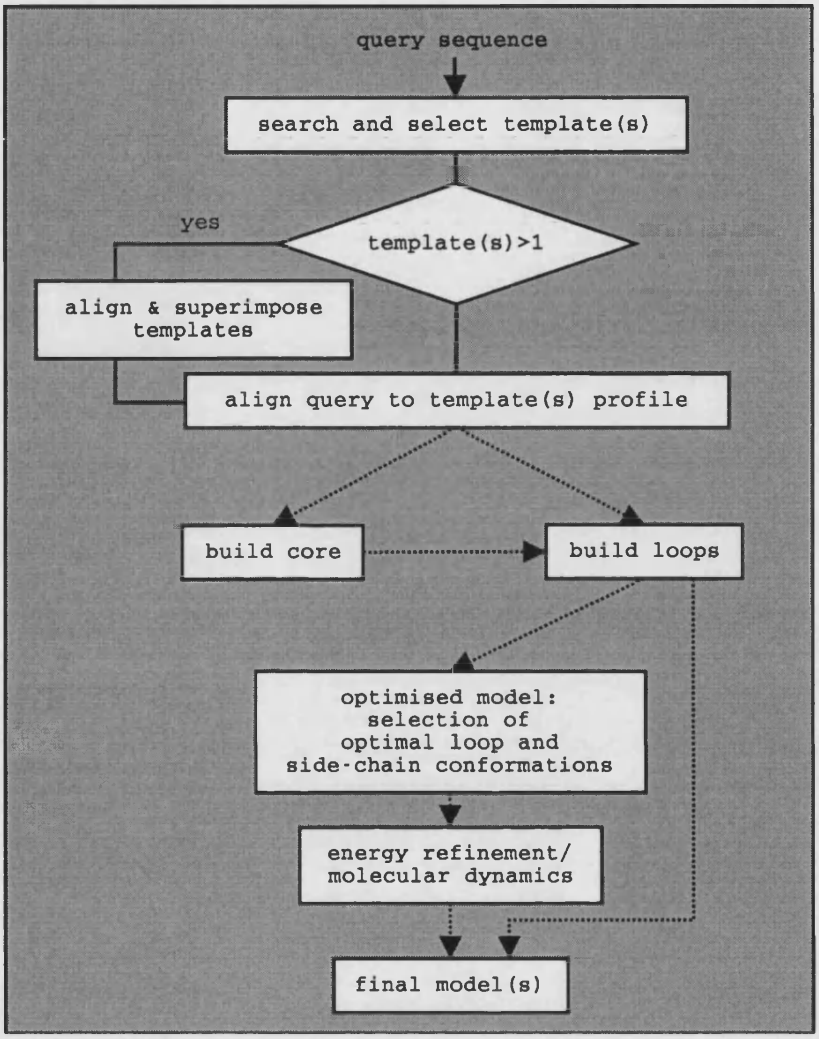


Figure 1.7: Generic steps in comparative modelling protocols. Dotted lines indicate optional or parallel steps. See also Table 1.1.

(Rost, 1995; Kelley *et al.*, 2000; Shi *et al.*, 2001), or by combining different alignment strategies (Elofsson, 2002). However, as low sequence similarity templates generally yield low accuracy models (Vitkup *et al.*, 2001), some comparative modelling programs, for example SWISS-MODEL (Guex *et al.*, 1999), use less ambitious and simpler methods to assure the quality of their results at the risk of missing some modelling targets.

Most of the above methods for identifying suitable templates perform local alignments, by finding maximum scoring sequence patches, which do not necessarily correspond to complete protein domains. For this reason databases of protein structural

domains, such as SCOP or CATH (see also 1.3.1), have been used to define templates (Kelley *et al.*, 2000). For the same reason multi-domain proteins remain a problem for comparative modelling programs and most modelling programs rely on the user's knowledge of how to split their query sequence into domains before submission.

1.4.2 Aligning the templates to the query

Once the complete set of possible template(s) have been found, it is necessary to select a subset from which to build the actual model. Modellers have long preferred to use several templates where available (Sali & Blundell, 1993; Guex *et al.*, 1999; Bates *et al.*, 2001; Venclovas, 2001), but the practical advantage of this approach has not yet been proven (Tramontano *et al.*, 2001). Indeed, most methods would perform better if the single ideal template could be recognised, but unfortunately pairwise sequence identity is not a consistent criterion by which to address this question (Wood & Pearson, 1999; Koehl & Levitt, 2002). If several templates are to be used they have to be optimally aligned to drive the process of model building. ClustalX (Thompson *et al.*, 1994), T-Coffee (Notredame *et al.*, 2000) and similar programs can be used for this, despite the fact that they can only produce approximations to optimal solutions for more than two sequences. But because sequence similarity between templates can be very low it may be necessary to use their structural similarity to align them. In this case programs such as SSAP (Taylor & Orengo, 1989), STAMP (Russell & Barton, 1992) or CE (Shindyalov & Bourne, 1998) may be used.

Finally, the query sequence needs to be accurately aligned to the template(s); again sequence and structural information is often used. Typically the alignment procedure must exclude gaps in secondary structure elements and anchor the alignment in non-loop regions. In addition, key functional motifs should also be correctly aligned, for example P-loops (Walker *et al.*, 1982), EF-calcium-binding loops (Kawasaki & Kretsinger, 1995) and catalytic triads(Branden & Tooze, 1999). Databases of such motifs have been constructed, including PRINTS (Attwood *et al.*, 1998) and BLOCKS (Henikoff *et al.*, 1999); however, I am unaware of any automatic modelling procedure that takes advantage of these useful sources of information.

1.4.3 Modelling by satisfaction of spatial restraints

This family of approaches was first proposed in the mid-eighties (Braun & Go, 1985; Havel & Snow, 1991; Sali & Blundell, 1993) and consists of computing geometrical and

biochemical restraints from the set of superimposed templates that the aligned query sequence will have to optimally satisfy. This method considers the possible templates as a sample of the folding space for a group of homologous proteins. Since the query sequence is believed to be another homologous member of the group, it will have to fulfill the restraints dictated by its relatives. As a consequence, models built using this method are derived from every template used and do not directly inherit backbone segments from any one template. Optimisation of possible conformations according to the restraints can be done in a variety of ways ranging from conjugate gradient minimisation (Sali & Blundell, 1993), simulated annealing (Ogata & Umeyama, 2000) and graph theory (Samudrala & Moult, 1998). The weakness of the method is that templates need to be reasonably superimposable to define the restraints and that some regions are poorly restrained. Its strength however, is that it can directly model an entire protein structure as a continuous chain. Methods which essentially apply distance constraints to reconstruct the protein backbone, such as neural networks (Lund et al 1997), also fall into this category.

1.4.4 Modelling by fragment building approaches

This has historically been the most popular approach for comparative modelling, which grafts protein fragments from the template(s) to build up the query structure (Greer, 1981; Jones & Thirup, 1986; Blundell *et al.*, 1987; Sutcliffe *et al.*, 1987a; Guex *et al.*, 1999; Bates *et al.*, 2001). This method has clear limitations in modelling sections which differ widely between templates, such as loops, because the matching of the selected fragments is non-trivial and often requires additional modelling steps (see below). However, the benefit of the approach is that sections confidently inherited from the templates have intrinsically good geometry and require minimum subsequent optimisation. A related but novel methodology has recently been applied to *ab initio* protein structure prediction. This uses small protein fragments extracted from templates which are not necessarily homologous (Unger *et al.*, 1989; Simons *et al.*, 1997a), allowing models to be built where no significant sequence similarity is found to any template. The accuracy limits for these methods has been recently benchmarked by Kolodny *et al.* (2002), with average backbone RMSDs ranging from 0.8Å to 2.9Å , depending on the fragment library used.

1.4.5 Optimisation: selection of side-chains and loops

Once the conserved core of the model has been constructed, most protocols then investigate loop and side-chain optimisation. In the context of a protein, a loop can be defined as a region of variable length and irregular shape connecting secondary structure elements

(Branden & Tooze, 1999) (see also Section 1.1.2). If there is a high sequence similarity with the template, then these homologous loops may be modelled in a similar way to the rest of the protein (Greer, 1981). The methods for constructing loops for less conserved regions fall into two main categories, database searches and *ab initio* methods.

Database searches are based on grouping observed loops in the PDB and building a library. The method relies on the assumption that the set of structures used is large enough to produce a database which cover all possible geometrical configurations that protein loops can adopt. However, as segments of up to nine residues with the same sequence can have completely unrelated conformations in different proteins (Sander & Schneider, 1991; Mezei, 1998), sequence alone can not be used as a method of defining useful groups. Early classification systems relied on manual investigation of loops within specific environments, such as β-turns (Ventkatachalam, 1968), γ-turns (Rose *et al.*, 1985; Milner-White & Poet, 1986) and α-α,α-β, β-α and β-β arches (Edwards *et al.*, 1987; Rice *et al.*, 1990; Colloc'h & Cohen, 1991; Efimov, 1991). More recently, automatic classification systems have been used, which include information about the structures flanking the loop and clustered based on RMSD (Kwasigroch *et al.*, 1996; Wintjens *et al.*, 1996). More specific and tighter clusters have also been generated by specifically taking into account bracing geometry, Ramachandran patterns and sequence (Oliva *et al.*, 1997).

Ab initio loop prediction methods are based on a conformational search of the space to be filled. There are many methods and different potential energy functions to discriminate between possible conformations, including systematic conformational searches, molecular dynamics simulations, Monte Carlo techniques, genetic algorithms and dynamic programming (see for example (Contreras-Moreira *et al.*, 2002) and articles referenced therein). It is not clear yet whether database or *ab initio* methods are the more accurate for small to medium size loop construction. For example, in 1994 a study looking at the effectiveness of database methods concluded that they were only sufficient for loops of up to 4 residues (Fidelis *et al.*, 1994). However, later work showed that with some optimisation of the loops the limit for databases searches could be raised to 9 residues (van Vlijmen & Karplus, 1997) - for a loop of this size *ab initio* methods need to generate substantial numbers of loop configurations to fully sample conformational space. What is clear is that in both database and *ab initio* methods, a scoring function is required to select the correct loop from the ensemble searched. Many scoring functions have been tried and the effectiveness of these dictates the final accuracy that can be attained. Scoring functions remain a problem and may require a deeper consideration of complete free energy summations that include appropriately weighted terms for example of loop entropy (Xiang *et al.*, 2002) and desolvation (Janardhan & Vajda, 1998).

Usually, the second phase in optimising a model is the addition and refinement of the side chains. Side-chain prediction algorithms almost exclusively use a database of rotamers, as this significantly reduces the complexity of refining all the side chains in a protein at the same time. Early work had noticed that there was a significant tendency for side chains to prefer certain rotameric states depending on secondary structure (McGregor *et al.*, 1987; Sutcliffe *et al.*, 1987b). Similar investigations led to the production of backbone dependent rotamer libraries (Dunbrack & Karplus, 1993; Bower *et al.*, 1997). Using these libraries, the simulated annealing method used by Lee & Subbiah (1991) was reasonably successful at predicting side-chains of the hydrophobic core of proteins. A significant reduction in the number of combinations of rotamers to search was made possible by the dead-end elimination method (Desmet *et al.*, 1992; Lasters & Desmet, 1993; De Maeyer *et al.*, 2000), which allows the early elimination of impossible rotameric combinations. Other methods for searching side-chain combinations were also developed, one of the most widely used being the self-consistent mean-field approach (Koehl & Delarue, 1994).

Many of these approaches are often tested on known crystal structures with the side chains removed. Whilst this is fine for checking the accuracy of the methods, it does not check the accuracy when used for predicting side-chain conformations for a comparative model which has backbone errors inherited from the modelling process. Desjarlais & Handel (1999) developed a method that allowed flexibility in the backbone at the same time as the selection of the side chains. This showed that even in core regions, significant changes to the backbone inherited from homologous proteins can occur to accommodate the new side chains, and current methods that do not include backbone flexibility would be severely impeded in choosing the correct rotamers. It was also assumed that core regions were exclusively dictated by van der Waals packing. However, this has been shown to be insufficient on its own to define these regions (Kussell *et al.*, 2001). Recent work (Xiang & Honig, 2001) has concluded that there is no combinatorial problem in the choice of the correct side chain on a correct backbone, but that as long as a highly detailed rotamer library is used the limiting factor becomes the scoring function. A detailed study (Jacobson *et al.*, 2002) into surface side chains has shown that the crystal environment has significant effect on the final conformation adopted. In addition, limits for the maximum accuracy were also calculated which showed that, whilst it should be possible to predict core regions to 90% accuracy compared with the X-ray structure, many surface side chains adopted many different conformations dependent on their environment. Therefore,

predicting single rotamer states for exposed side chains is not justified. Given these constraints, many modern methods do manage to achieve a reasonable level of accuracy and even reach the limit in the core regions (Mendes *et al.*, 1999; Petrella & Karplus, 2001; Liang & Grishin, 2002).

1.4.6 Energy refinement and molecular dynamics

As a final step some form of energy refinement is usually performed on the models. This can be achieved by using one of the energy minimisation software packages such as CHARMM (Brooks *et al.*, 1983) (see 1.3.4). This step requires adding covalent hydrogen atoms, generally ignored during the construction of the model. In addition, depending on the pH and the local environment, the protonation state of basic and acid groups may change. Unless specific environments are to be studied, generally a neutral pH is assumed. If cofactors and their positions are known, they should be added, although only the most common ones are currently included in force-fields.

Refinements obtained with these approaches usually have a small radius of convergence and are used simply to remove steric clashes, particularly between side chains, and ensure sensible covalent geometry is maintained around each atom. Often this achieves little more than improving the appearance of the model (Schonbrun *et al.*, 2002). Indeed, there has been little work done to show if energy refinement does in general slightly refine models in the correct direction. A technique that enables a larger radius of convergence, compared to standard energy minimisation, is molecular dynamics. However, in a recent study on a small number of protein models using state-of-the-art explicit solvent molecular dynamics, only limited success was achieved in refining some of the models closer to the native state (Lee *et al.*, 2001).

1.4.7 Error analysis

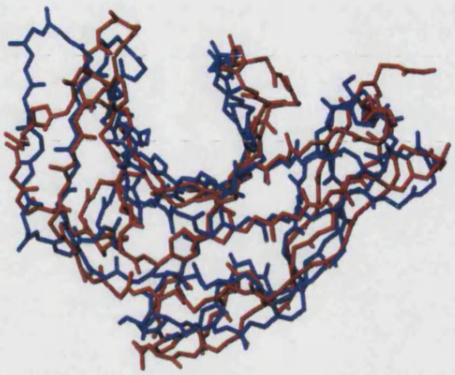
What are the most common errors in comparative models? Previous work (Marti-Renom *et al.*, 2000; Bates *et al.*, 2001; Tramontano *et al.*, 2001) identifies three major sources of errors in comparative models: template selection, sequence alignment and loop/side-chain building. Selecting templates becomes especially difficult when their sequence similarity to the query is low (less than 25-30% of sequence identity). In these circumstances even statistically significant sequence matches, for example found by BLAST, can identify totally different folds. As explained in detail above, there are many different sequence alignment methods but so far none can be considered optimal. However, whilst sequence identity is not a consistent measure of expected alignment accuracy (Tramontano *et al.*,

2001), alignments over 40% of sequence identity between query and template can be considered confident (Marti-Renom *et al.*, 2000). Below this threshold, alignments tend to accumulate errors. Unfortunately, these errors are inherited by the rest of the modelling process and current protocols are not able to detect them. A possible solution to this has been investigated by building models from several alternative alignments and then choosing the best based upon energetic or statistical potentials (Melo *et al.*, 2002). Finally, whilst no method is perfect, it has been shown that by using several protocols the optimal alignment may be obtained - the problem is then reduced to being able to routinely identify this alignment (Elofsson, 2002).

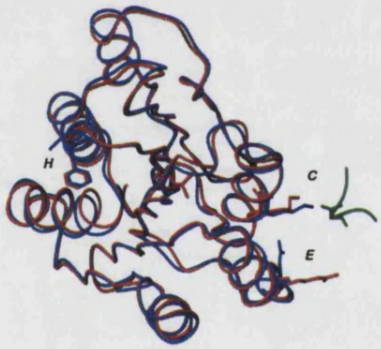
Even in confident regions of sequence similarity quite different backbone conformations can be present in a comparative model compared to the native or target structure. These can confuse rational experimental design and occur essentially because proteins are flexible (see Figure 1.8 A) - proteins can exhibit different conformations depending on their environment (Branden & Tooze, 1999; Liu *et al.*, 2002). A clear example of this problem is seen in globular proteins that build the 30S ribosome. Many of them have been solved independently and as part of the ribosome and they show important differences in exposed loops and N- and C-termini that seem to be important for function (Brodersen *et al.*, 2002). If these structures are used as templates they will yield different models for the same protein.

If we are sure that the above alignment problems do not affect the model under construction we can then consider loop building errors as the next major problem. Loops can be confidently modelled if they are only up to 5-6 residues long (Martin *et al.*, 1997). In fact, as described above, loops of this size tend to form conformational clusters (Oliva *et al.*, 1997; Branden & Tooze, 1999). Longer flexible fragments are usually not accurately modelled and indeed some modelling protocols simply do not attempt to model these regions (Venclovas, 2001). However, since loops are frequently important for protein function (Oliva *et al.*, 1997) and are sometimes difficult to detect even for X-ray or NMR structure determination experiments, we must look further for solutions to this essentially mini protein folding problem. One possible solution to this could be to consider an ensemble of low energy loop conformations within a broad free energy minimum (Xiang *et al.*, 2002).

The next level of uncertainty in models is at the side-chain level. As discussed above, provided the modelled backbone quality is high, side-chain are usually well placed in the protein core but are subject to variations at the surface, as shown in Figure 1.8B. The uncertainty in surface side-chain rotamers can sometimes be resolved when considering protein-protein interactions as these reduce their degree of flexibility.



(a)



(b)

Figure 1.8: **(a)** An example from the automatic server 3D-JIGSAW, showing a backbone model (blue), based upon an NMR template, superimposed on the high resolution structure of the same protein eventually solved by X-ray crystallography (red). NMR (template) and X-ray structures have identical sequences. Interestingly, there are many conformational differences throughout the fold (not just loop regions) giving a final backbone RMSD of 2.5Å. **(b)** Cartoon of a model (red) showing minor deviations from the experimental X-ray structure (blue) modelled with 3D-JIGSAW from a 95% identical template. Hydrophobic core side chains (marked with *H*) agree well with the observed; however, exposed side chains (*C*) can show significant differences in their rotameric states due to crystal contacts (indicated here by the green chain), or simply because they are exposed to solvent (*E*), suggesting they may have multiple rotameric states.

Finally, a common problem in comparative modelling is to calculate exact relative

domain orientations in multi-domain proteins. Surprisingly, given the large RMSD errors involved, this appears to be a subject for which a comprehensive study has not as yet been performed. Molecular dynamics and protein docking techniques may aid the solution to this domain-packing problem (see for example (Janin *et al.*, 2003)).

1.4.8 Quality control

What kind of RMSDs are we likely to expect between model and the experimentally determined structure? Chothia & Lesk (1986) studied the sequence and structural variability within protein families and observed that as the sequence similarity between proteins decreased, the RMSDs between their superimposed structures increased exponentially. Based on the results from CASP experiments (see Section 1.5), similar studies have been conducted on protein model quality relative to closest template (Vitkup *et al.*, 2001). Figure 1.9 shows the latest results from the EVA experiment (Eyrich *et al.*, 2001) (see Section 1.5) plus our own in-house benchmark of model accuracy. In general, regardless of the servers used, for proteins sequences around 95% identical the backbone RMSD is expected to be under 1Å; when the sequence identity drops to 30%, the expected RMSD is around 4Å. As can be seen in the figure, there is an increasing range of variability around these error estimates towards lower sequence identities.

Apart from the grosser limitations to the use of protein models dictated by sequence similarity to the templates, the user can check the stereochemical and thermodynamical quality of models by using programs such as PROCHECK (Laskowski *et al.*, 1993) and WHATCHECK (Hooft *et al.*, 1996). Another way to validate comparative models is to check whether the implications of the modelled structure agree with experimental observations, such as mutations or biochemical measures, or observations found in the literature.

However, until a rigorous ranking scheme for model accuracy can be found, the final indication of the correctness of a model protein will always lie in the hands of the experimentalist.

1.4.9 Applications of CM

As a consequence of the above quality controls it is possible to enumerate the applications for which protein models are likely to be useful according to the sequence identity between query and template (Marti-Renom *et al.*, 2000; Baker & Sali, 2001). Traditionally, molecular biologists have used protein models to design site-directed mutagenesis, engineering experiments or to understand mutant phenotypes in the light of protein structure.

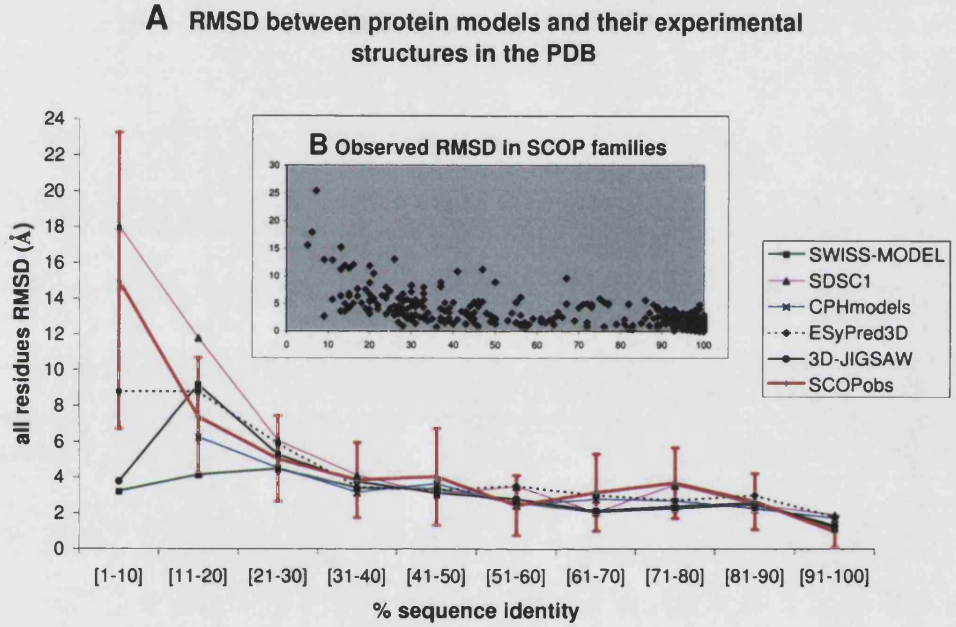


Figure 1.9: **A**, comparison of observed accuracy for models returned to the assessors for the EVA experiment. $C\alpha$ RMSDs are reported versus % sequence identity to the closest template. There are five server results plotted, indicated by the first five labels in the figure key (see Table 1.1 for more details), plus a benchmark plot from pairs of SCOP family members (SCOPobs). The error bars show the extent of variation expected for each sequence identity sub-group (binned every 10%). **B**, (axis titles as in **A**), individual observations in the plot of pairwise SCOP families used in the calculation of error bars for **A**.
This Figure was prepared with help from Paul W.Fitzjohn.

Even very low sequence identity templates yield useful models some of which have given insights into potential protein functions, see for example (Garmendia *et al.*, 2001; Devos *et al.*, 2002). Apart from functional study applications, low-resolution models are also being used to build supra-molecular structures (Zhang *et al.*, 2000; Wriggers & Chacon, 2001; Aloy *et al.*, 2002; Elcock, 2002). Mid-resolution models, derived from templates around 50-60% identity level, could be valuable as models for use in molecular replacement (X-ray crystallography) and the rational design of more stable proteins, for example the addition of a disulphide bond (Mansfeld *et al.*, 1997). Finally, high resolution models, those obtained typically from templates over 90% identical in sequence, are being used routinely as receptors to dock and rank small molecules for potential pharmaceutical use (Mangoni *et al.*, 1999; Schafferhans & Klebe, 2001; Peitsch, 2002). In addition, it is accepted that the growing interest in unveiling protein-protein interactions can benefit from the contributions of comparative modelling and docking programs (Tovchigrechko *et al.*, 2002).

In terms of finding disease-related proteins and for preliminary investigations of potential drugs to modulate the functions of these proteins, the most important genome to generate complete three-dimensional models for is obviously our own - the human genome. Figure 1.10 shows the number of human proteins with at least one domain that can be modelled using comparative modelling techniques. We estimate that up to 38% of the translated genome contains domains which can be modelled using templates of at least 20% sequence identity. This would mean an expected accuracy for the conserved core of each model between 0.9 and 4.0Å $C\alpha$ RMSD. These models could be used for any of the tasks mentioned above or to understand the structural effects on proteins due to single nucleotide polymorphisms (Wang & Moult, 2001) or genetically characterised diseases at the molecular level (Hogg & Bates, 2000; Huyton *et al.*, 2000; Sellar *et al.*, 2003).

1.4.10 Problems and potential solutions

As experiments like CASP have shown, comparative modelling involving some form of human intervention still produces models of higher quality than models produced from completely automatic procedures. Intervention seems to be particularly critical in selecting adequate templates and tweaking the alignments (Bates *et al.*, 2001; Venclovas, 2001). Therefore, more algorithmic development is required if we are to automatically select optimal templates and alignments. Some progress has recently been made with the former problem by selecting templates from large ensembles of sequences, theoretically generated according to their structural compatibility with a template (Koehl & Levitt, 2002).

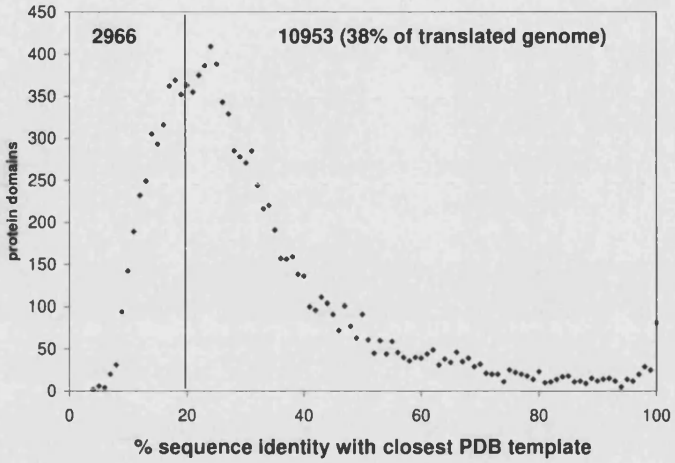


Figure 1.10: Distribution of human proteins containing at least one domain with significant sequence similarity to SCOP domains. The vertical bar separates the fraction that can be modelled to at least a level of resolution that may be useful for experimental design such as site-directed mutagenesis. Over half of the human genome (proteins not represented in the plot) cannot confidently be assigned to known protein folds. These assignments were made using 3D-GENOMICS (Muller *et al.*, 2002).

Recently the latter problem has also been addressed by a consideration of a weighted contribution of a number of current sequence alignment protocols (Elofsson, 2002) (see also 3.7).

Irrespective of the above problems, increasingly more is being asked of comparative modellers. For example at CASP4 they were expected to model as low as 13% sequence identity with the closest template, and for CASP5 (held during the summer of 2002), of the 38 targets considered to be within reach of comparative modelling, 10 have only between 10 and 20% similarity to the closest template. Many of the algorithms designed for comparative modelling were not specifically designed to model at these very remote levels, as this was then considered more the domain of fold recognition experts. Interestingly, this is leading to a progressive merging of the fold recognition and comparative modelling fields. Comparative modellers are learning from the fold recognition community how best to detect very remote sequence relationships and how best to align the query structure to those templates once identified. Equally, those in the fold recognition community are learning how to generate full three-dimensional models from their fold recognition and alignment algorithms. Hopefully this will create a second generation of algorithms, or a blend of algorithms, that are much more likely to be successful across

a wide range of sequence similarity between query and template sequences. Together with this convergence of algorithms, and on the assumption that only a limited number of protein folds exist, rational structural genomics efforts may be the key to allow three-dimensional modelling of any sequence in a matter of years (Baker & Sali, 2001; Vitkup *et al.*, 2001). However, the end-game of protein modelling, refining medium resolution models to high levels of atomic accuracy, levels of accuracy routinely obtained in X-ray structures, may take considerably longer as more sophisticated force fields (Halgren & Damm, 2001) (see also 1.3.4) and substantially more computer power at the fingertips of developers may be required.

1.4.11 Web-based modelling

Although there are a number of well maintained downloadable comparative modelling software packages available, the future of comparative modelling as an essential tool for biologists may be the growing number of web-based servers. Table 1.1 summarises the tools that are currently freely available for academic use. The advantage of web tools is that they are easy to run, even across different computer platforms, often only requiring the query sequence and user's e-mail address. In addition the sequence and structural databases that the algorithms require are usually maintained by the developer; thus linking software to the appropriate up to date databases is not a problem. Some of these servers are now allowing some user intervention in the model building process, for example SWISS-MODEL allows choice of templates and our own server, 3D-JIGSAW, allows both template selection and manual adjustments of the query to template alignments.

Server/program name	URL/Modelling method
3D-JIGSAW	http://www.bmm.icnet.uk/servers/3djigsaw Looks for homologous templates and splits the query sequence into domains. If good templates are found the best covered domains are modelled, currently using a maximum of two templates. Different loops are tried to connect secondary structure elements taken from the templates. The best ensemble is then refined (Bates *et al.*, 2001; Contreras-Moreira & Bates, 2002).
CPHmodels	http://www.cbs.dtu.dk/services/CPHmodels A neural network based method to predict C-α contacts to drive the sequence alignment. No side-chains are constructed (Lund *et al.*, 1997).

continued on next page

continued from previous page

Server/program name	URL/Modelling method
EsyPred3D	http://www.fundp.ac.be/urbm/bioinfo/esypred Exploits a new alignment strategy using neural networks. Complete models built with MODELLER (Lambert *et al.*, 2002).
FAMS	http://physchem.pharm.kitasato-u.ac.jp/FAMS Templates found by sequence similarity are superimposed to define the structural landscape of each residue in the query sequence (similar ideas to MODELLER). Protein fragments with their side-chains are then sampled to fit the observed landscape using a simulated annealing algorithm (Ogata & Umeyama, 2000).
Nest*	http://trantor.bioc.columbia.edu/~xiang/jackal Allows building models with one or several templates tuning their alignments and permitting artificial evolution.
MODELLER*	http://salilab.org/modeller/modeller.html Builds a complete model based on alignments prepared by the user. The procedure is based on satisfying spatial restraints (automatically computed from the templates used). Models are refined using a variety of algorithms (Sali & Blundell, 1993; Fiser *et al.*, 2000).
ModzingerZ	http://peyo.ulb.ac.be/mz Templates are aligned to the query sequence to build a library of backbone fragments. Fragments are then combined to build alternate models and scored. Finally side-chains are added.
PCOMB	http://www.sbc.su.se/~arne/pcomb Pcomb uses a combination of several sequence-profile and profile-sequence searches. Final models are produced using MODELLER.
PROTINFO	http://protinfo.compbio.washington.edu A core model is built for each template found by sequence similarity to the query. Loops and side-chains are then added to the best scoring models.
SWISS-MODEL	http://www.expasy.ch/swissmod BLAST found templates are multiply superimposed and then aligned to the query sequence excluding loop regions. The core is then calculated as a weighted average of the templates. Loops are then added and the final model is refined (Guex *et al.*, 1999).

continued on next page

continued from previous page

Server/program name	URL/Modelling method
TSUNAMI	http://www.pirx.com/tsunami Fragments of templates found by a BLAST-like algorithm are assembled and the final model is evaluated using statistical potentials.

Table 1.1: Freely available CM web-servers and programs. These programs return atomic coordinates to the user. Most fold-recognition servers return only alignments and therefore are not listed here. (* indicates downloadable software)

1.5 CASP blind trials, EVA and LiveBench

There is a formal quality control procedure to test and evaluate new prediction techniques every two years, the Critical Assessment of techniques for protein Structure Prediction (CASP) experiments. As the number of protein structures predicted in each CASP experiment has been small the statistical significance of ranking the prediction methods has been brought into question (Marti-Renom *et al.*, 2002). However, the value of human expert analysis should not be underestimated as developers gain additional insights into further developing their algorithms beyond that given by pure numerical analysis. For example, advantageous ways to mix current algorithms may be suggested.

To address the statistical weakness of CASP and to help modellers test their algorithms on a more frequent basis, two continuous assessment projects have recently started: EVA (Eyrich *et al.*, 2001) and LiveBench (Bujnicki *et al.*, 2001), more focused on fold recognition programs. In these experiments, sequences of proteins about to be released in the PDB database (determined experimentally) are automatically sent to participant servers around the world, which in turn send back automatically built protein models. The benefit of such on-line experiments is that the evaluation of model quality is also fully automatic and so the results for each server in the experiment can be posted on the web very quickly and at regular intervals; EVA results for example are tabulated weekly. This enables molecular biologists to determine which server(s) are currently likely to give them the more accurate models and helps developers rapidly benchmark and rank their new modelling algorithms against others in the field. The handicap of these methods is that although an extensive numerical analysis is performed there is no human overview of the interplay between these results and the variety of complex methods used to obtain them.

1.6 Structural Genomics

As a complement to genome sequencing projects, the Structural Genomic initiative (and participating projects around the world (see http://www.structuralgenomics.org/)) have the goal of obtaining useful three-dimensional models of all known proteins by a combination of experimental structure determination and CM (Vitkup *et al.*, 2001), or more generally, by combining different and complementary experimental and theoretical techniques (Sali *et al.*, 2003). The idea is to optimize the efforts in such a way that a minimum number of experimental structures can be used to maximize the application of CM (and other techniques) to the remaining proteins. For this to be achieved, experimental targets must be selected according to the distribution of their homologous sequences. Estimates by Vitkup *et al.* (2001) suggest up to 16,000 structures may be needed. This calculation does not account for membrane proteins, usually excluded from these initiatives, or proteins technically difficult to study with current experimental procedures, for example proteins containing highly flexible or low complexity regions (Liu *et al.*, 2002), and assumes that CM is reliable provided that the sequence identity between solved structures and the remaining is of at least 30%. To coordinate efforts, list of targets waiting to be solved are regularly updated, as well as proteins currently under experimental study (see for instance http://www.jcsg.org).

Building this comprehensive set of protein structures (known and predicted) is expected to be beneficial in a number of ways (Burley, 2000):

- Each one of these structures can serve as a starting point for a rational program of experimentation, such as site-directed mutagenesis, ligand binding studies, enzyme assays or protein-protein interaction studies.
- Should the structure represent a new fold with a known function, it may well be possible to identify regions of the protein responsible for function *in silico* by comparing the newly determined structure with those of structurally distinct yet functionally similar proteins.
- When the structure proves to be a known fold with a known function, we can expect to learn more about divergent/convergent evolution. This has been the case for many TIM barrel enzymes, which catalyze a wide variety of chemical reactions using the same protein fold decorated with different patterns of surface-accessible residues creating functionally distinct active sites.

- Where we do not know anything about biochemical function, both new and previously known structures should still prove useful. The newly determined structures that are not in fact novel can be compared with their structural homologs, and it may be possible to infer function.

- If a new fold is found with no functional information available at all, it may be characterized by scanning it against a library of all known binding sites and enzyme active sites.

There are however some limitations for these Structural Genomics projects. The main problem is the fact that the structure of an isolated protein may not indicate its biological function(s) if it normally resides in a macromolecular complex. In addition, the so-called low complexity regions, which may never adopt stable conformations or remain unstructured until they interact with their respective targets (Liu *et al.*, 2002), are clearly beyond the initial scope of structural genomics projects(Burley, 2000).

1.7 Outline of thesis

Comparative modelling is a widely used methodology in Structural Biology. Several problems affect the performance of CM and therefore solutions for them are needed to increase its applicability. Here, different algorithms and approaches were tested with this aim and some useful insights were obtained. Interestingly, a biologically inspired algorithm described in Chapter 3 guided us to find some evolutionary features of protein families which may have implications for protein design.

The main aspects of this work are now introduced.

- Chapter 2 describes the development of a tool to split protein sequences according to structural domains and to align available templates to each sequence segment. The web server DomainFishing was later created to implement these ideas. The performance is benchmarked and some problems are identified.

- Chapter 3 describes a theoretical approach to recombine protein structures *in silico* as a different way to build comparative models. The method is extensively benchmarked both in the laboratory and also during CASP5 blind tests. Interestingly, the method is also useful for Fold Recognition.

- In Chapter 4 we explore the possible connections between gene structure and its corresponding protein fold by doing statistical analysis and some artificial recombination experiments. The data obtained suggests that the distribution of introns in

genes is sensitive to protein structure and that protein recombination experiments may reveal evolutionary features of protein families. In particular, a weak spatial correlation is found to those places in primary sequence where introns are less likely to occur. Some implications for protein design and related work are discussed.

- This thesis closes with a summary of the results and some concluding remarks, in Chapter 5. Possible improvements and suggestions for future work are included in every chapter.

Chapter 2

Alignments and templates in Comparative Modelling

2.1 The alignment problem

As soon as protein sequencing techniques, pioneered by Sanger (1952), were developed and sequences started to become available, the need for tools to compare them became obvious. The adopted way to compare protein sequences was to align them, as shown in Figure 2.1. Aligning proteins has many possible applications; three related to this work are highlighted here:

- Finding evidence for homology, the existence of a possible common ancestor relating the compared proteins and their genes.
- Inferring evolutionary constraints that may indicate the biochemical function, such as conserved binding sites or residues composing the hydrophobic core of proteins.
- Predicting secondary structure, useful for example in improving difficult sequence alignments or to select possible epitopes in proteins susceptible to be recognised by antibodies.

The elementary sequence alignment involves a pair of proteins. As suggested by Figure 2.1, alignment methods should have well defined metrics to score matching residues and should also be able to manage insertions and deletions in the primary structure. The most important algorithm developed for this purpose is that by Needleman & Wunsch (1970) to globally align two proteins a and b(or DNA sequences) of length n and m. This dynamic programming method aligns n and m from the first to the last residue maximising

Figure 2.1: Sequence alignment of three example proteins as depicted by the program ClustalX (Thompson *et al.*, 1994). Conserved columns are marked with $\star$. Note that in protein 3 an otherwise conserved Threonine residue (T) is substituted by Serine (S). A deletion (.) is shown in protein 2. The colour of each column usually describes biochemical properties of the residues highlighted. *Note that O does not correspond to any biological amino acid.*

the alignment score while allowing deletions in either sequence to occur. This method, as modified by Sellers (1974), uses two cost parameters for opening and extending a deletion (or gap) and also a precomputed 20×20 matrix containing the observed natural amino acid substitution frequencies (see Section 2.1.1). With this data, the procedure calculates $n \times m$ $D_{i,j}$ cumulative partial scores usually starting from the top left corner of the dynamic programming $n \times m$ matrix D towards the bottom right corner, following these simple rules (c_k is the penalty for introducing a gap of length k):

$$D(i,j) = max \begin{cases} D(i-1,j) - c_k & \text{deletion at position j (cell above)} \\ D(i-1,j-1) + score(a_i,b_j) & \text{substitution } a_i,b_j \\ D(i,j-1) - c_k & \text{insertion at position j (cell to the left)} \\ 0 & \textit{(local alignments, if } D(i,j) < 0\textit{)} \end{cases} \tag{2.1}$$

These rules imply that an extra column and and extra row are needed to start the calculations, as frames. In the original global alignment algorithm, these extra elements in the matrix are filled using a linear function of the gap cost, starting on the top left corner. When the matrix has been filled, the cell containing the maximum value along the bottom and right borders is chosen as the alignment start and a trace back route is found until the left or top borders are met, maximising the overall score. Smith & Waterman (1981) modified this approach to make local alignments. Here the starting cell for the alignment, the maximum, is searched in the whole dynamic programming matrix and the trace back is stopped as soon as a cell containing a 0 score is found.

These alignment algorithms are relatively expensive as they require $n \times m$ calculations and so their time complexity is the product of the length of the sequences involved. Due

to the large size of current sequence databases, these algorithms are usually not used to search them. Instead, faster algorithms that explore only relatively minor fractions of the alignment space encompassed by a pair of proteins are preferred. In Section 2.1.2 one of the most successful methods of this type, BLAST, is described.

2.1.1 Scoring matrices

A substitution scoring matrix is a 20×20 matrix S in which each cell contains an empirical value to score the substitution of one natural amino acid by another one. These matrices are tailored so that they reproduce a desired behaviour, such as producing alignments similar to those that an expert would do, or aligning residues in a way that agrees with observations in nature. A general equation for substitution scores S_{ij} would be:

$$S_{ij} = \frac{log\frac{q_{ij}}{p_i p_j}}{\lambda} \tag{2.2}$$

where q_{ij} is the observed exchange frequency with which amino acid i is replaced by amino acid j, as observed when analysing natural mutations in groups of clearly homologous proteins. p_i and p_j are background frequencies of residues i and j in all known protein sequences. S_{ij} is then multiplied by a factor and rounded to the nearest integer for simplicity. These scores are denominated log-odds and may be divided by a scaling factor λ, specific for each scoring system (for BLAST using a BLOSUM62 matrix, takes the value 0.267). Most scoring matrices assume that the expected score S_{ij} for a chance amino acid substitution in a comparison of two random sequences would be negative. The explanation for this is that, otherwise, random alignments would have positive scores if long enough.

The most common generic scoring matrices are PAM and BLOSUM. The choice of the substitution matrix, and in general the scoring scheme used, is crucial for the quality of the alignments obtained, but no single scoring system appears to be the best for all purposes (Elofsson, 2002). These matrices are of general use and therefore can, in principle, be applied to many different proteins. However, more specific matrices would perhaps allow better alignments for specific problems.

PAM matrices

The Point Accepted Mutation (PAM) matrix models the evolutionary distance between sequences of closely related proteins (Schwartz & Dayhoff, 1978). Cells in the matrix contain the estimated probability of exchanging amino acid i with residue j after a given

evolutionary interval measured in PAM units. One PAM is the probability of a residue to be mutated during an evolutionary distance in which 1 in 100 point mutations was accepted. The original PAM250 matrix was based on a database of 1572 changes in 71 groups of closely related proteins. PAM matrices for longer evolutionary distances can be obtained by multiplying each target exchange frequency of the PAM1 matrix n times with itself to generate a PAMn matrix. By trial and error Schwartz & Dayhoff (1978) found that a PAM 250 matrix works well for distant relationships. The main problem with this PAM measure of distance is that it assumes that all positions along a protein sequence are equally mutable, and that is clearly not the case.

BLOSUM matrices

BLOSUM matrices were derived from conserved sequence blocks obtained from the BLOCKS database (Henikoff & Henikoff, 1992; Henikoff *et al.*, 1999). Frequencies of amino acids in these blocks of homologous sequences were tabulated and exchange and background probabilities calculated. Each block is a cluster of proteins built using a minimum % of sequence identity, n. The most common matrices are BLOSUM50, BLOSUM62 and BLOSUM80, where the number indicates the n% cut-off. BLOSUM matrices are constructed from sequences of any evolutionary distance without theoretical extrapolation, in contrast to PAM matrices. BLOSUM62 is shown in Table 2.1.

Gonnet matrices

A different method to measure differences among amino acids was developed by Gonnet *et al.* (1992) using exhaustive pairwise alignments of proteins from the MIPS protein sequence database (Mewes, 1991). They used PAM distance matrices to calculate initial alignments. These alignments are subsequently used to recalculate new scoring matrices. This process is iterated until convergence. The obtained scoring matrix is the Gonnet250 matrix and according to their results it should be used in preference to a PAM250 matrix.

2.1.2 BLAST, PSI-BLAST and IMPALA

As protein databases grew, several heuristic methods to speed up sequence searches were developed. Here the BLAST(Basic Local Alignment Search Tool) (Altschul *et al.*, 1990) method and its derivatives PSI-BLAST(Altschul *et al.*, 1997) and IMPALA(Schaffer *et al.*, 1999) are described since they have been applied extensively in this work. The basic principle is that significant sequence similarity may be found by comparing short protein

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

Table 2.1: The BLOSUM62 scoring matrix

fragments without dynamic programming. If short fragments from two proteins match, more sensitive and time consuming refinement steps can then be applied (including dynamic programming). These methods do not guarantee optimal alignments between two sequences but allow large databases to be scanned in a practical period of time. The BLAST algorithm to align two sequences includes five steps:

1. Find pairs of words of a given length (usually 3 residues for proteins) for which the cumulative score is at least T. A word satisfying this condition is called a hit. Scores are taken from a standard matrix such as BLOSUM or PAM. In a real scenario, all the possible words of the protein database are precomputed.

2. If at least two non-overlapping hits within a distance A are found on the same diagonal then the extension of these matches is triggered. If two hits overlap, the most recent one is ignored. Extending hits is the most consuming part of the algorithm.

3. The second hit is bidirectionally extended with no gaps until its cumulative score cannot be improved anymore. The extended hit may include other hits and is called HSP (High scoring Segment Pair).

4. The highest scoring HSP with a score $\geqslant S_g$, a predefined threshold, is further extended in both directions via a gapped alignment. Only the top scoring HSP is extended because most of the other HSPs will be included in it.

5. Final alignments for hits for which a gapped extension produced high scores are re-aligned with relaxed alignment parameters to be further extended. The final S score, the overall alignment score, is calculated.

Another scoring system is necessary in order to discriminate between meaningful and chance alignments. The distribution of ungapped local alignment scores for hits between a real protein sequence and a set of randomly generated sequences has been shown to follow a extreme value function (Karlin & Altschul, 1990, 1993; Altschul & Gish, 1996). Once the right set of parameters to describe this distribution is found, probabilities can be assigned to new hits according to the probability density function. In other words, the confidence of a sequence alignment to be meaningful can be measured as the probability to find at least one random alignment with score x. This probability is also known as a P-value and is calculated with equation 2.3, where K is a parameter that depends on the size of the search space and mn is the product of the lengths of the sequences that are

compared. λ is the same parameter as in equation 2.2.

$$P(S \geqslant x) = 1 - e^{-Kmne^{-\lambda x}} \tag{2.3}$$

The score S can be normalised as shown in equation 2.4:

$$S' = \frac{\lambda S - lnK}{ln2} \text{ (in bits)} \tag{2.4}$$

The reliability of an alignment in BLAST and similar programs is usually given as an *e*-value, as described in equation 2.5. This is the number of expected chance hits with a score $\geq S'$.

$$e(S') = mn2^{-S'} \tag{2.5}$$

This statistical framework has not been mathematically proven to work with gapped alignments, but computer simulations suggest it is still valid (Karlin & Altschul, 1990, 1993; Altschul & Gish, 1996). The main difference is that in the later case λ and K cannot be derived analytically and are therefore empirically approximated.

PSI-BLAST stands for Position Specific Iterative BLAST. Briefly, it is a iterative version of BLAST in which a standard scoring matrix is used only for the first iteration and subsequent iterations work using newly created matrices based on the confident hits found in the previous round (Altschul *et al.*, 1997). In particular, a position specific scoring matrix (PSSM) is created after every iteration to make the search increasingly specific to the family of sequences similar to the query. PSSMs are $n \times 20$ matrices where n is the length of the query sequence. This way, substituting an Alanine residue in position 23 may well have different costs than exchanging an Alanine in position 45. The use of PSSMs allows PSI-BLAST to be more sensitive than BLAST in finding remote homologous sequences (Altschul *et al.*, 1997; Park *et al.*, 1998).

The IMPALA computer program (Schaffer *et al.*, 1999) scans a query sequence against a library of PSSMs produced by PSI-BLAST using the Smith-Waterman (Smith & Waterman, 1981) method. IMPALA performs similarly to PSI-BLAST in terms of sensitivity and error rate (Schaffer *et al.*, 1999).

2.1.3 From pairwise to multiple alignments

If $n \geq 2$ proteins are to be simultaneously aligned we need a multiple alignment procedure. The described dynamic programming tools (see Section 2.1) are too expensive in

terms of computing time to be applied in the natural way, by creating a dynamic programming matrix of n dimensions. The time and space complexity of this approach scales up asymptotically to the order of l^n, where l is the average length of the sequences involved. Approximations are then required to generate multiple alignments. A common practical approximation is to build the multiple alignment in a progressive or hierarchical manner (Feng & Doolittle, 1987; Waterman, 1995). Instead of aligning all the sequences simultaneously, all-against-all pairwise alignments are first calculated to rank or cluster the n proteins in a hierarchical manner. Then these clusters of size ≥ 1 are solved independently and finally they are stacked according to the original ranking on a pairwise manner. In general, these methods calculate PSSMs for each cluster to score the corresponding pairwise alignment between clusters. After two clusters have been merged, a new PSSM is computed. Clustalw, probably the most popular multiple sequence alignment program, follows this clustering strategy based on a hierarchical guide tree(Thompson *et al.*, 1994). There are many variants of these methods, using sophisticated phylogenetic trees to guide the clustering or weighting clusters according to their content, but they have hardly been used for this work.

PSI-BLAST (see Section 2.1.2) generates its PSSMs in a different way, by piling up all the confident hits overlapping the query sequence and counting the mutations and their frequencies. From this point of view, PSI-BLAST PSSMs are not generated from multiple alignments, but from stacked significant hits.

A different approach to build sequence profiles are Hidden Markov Models (HMM), which associates different states (members of multiple sequence alignment and their residues) and the transitions between these with some probabilities. HMM based methods have not been directly used in this work.

2.2 Analysis of some alignment techniques in Comparative Modelling

As mentioned in Section 1.4.7, sequence alignment errors are critical for the quality of generated CM models. Therefore part of this project was dedicated to study this problem, implement alignment algorithms and test them. Comparative Modelling relies on the observation that homologous proteins have similar structure, with differences proportional to the degree of their amino acid sequence identity (Chothia & Lesk, 1986), as can be seen in Figure 1.9. This means that model accuracy is, at least with current alignment methods, highly dependent on the sequence similarity between query and template. Furthermore,

this logarithmic trend is not fully consistent, excellent models can often be constructed with very remote templates, or relatively bad ones based on very close homologous proteins. The initial goal we had in mind was to progress in the following directions:

i Improve sequence alignment procedures or at least learn from them.

ii Implement a reasonable measure of reliability and quality for our models based on the alignment to the template.

A way to evaluate alignment procedures is to produce a set of alignments and then compare them to alignments that are supposed to be correct. In this work, a correct alignment should be one that faithfully represents a structural superimposition of protein structures, one that matches residues in the partner sequence and that are neighbours in Cartesian space. Other definitions for the correctness of alignments are possible, such as the correct pairing of functionally important residues, but this is also expected to happen if the previous criterium holds. Since the alignment sets how spatial coordinates from the template would be adopted in the final model, this standard was adopted. Table 2.2 and Figure 2.2 show two complementary views for the same alignment, at the sequence level and in Cartesian space.

```
1d5ya  FKIETTPESRYLAQIGDSVSLTCSTTGCESPFFSWRTQIDSPLNGK
1bowa  -QTSVSP-SKVILPRGGSVLVTCSTSCDQPKLLGIET----PLPKK

1d5ya  VT--NEGTTSTLTMNPVSFGNEHSYLCTATCESRKLEKGIQVEIYS
1bowa  ELLLPGNNRKVYELS--NVQEDSQPMCYSNCPDGQSTAKTFLTV--
```

Table 2.2: Sequence alignment between human adhesion molecules ICAM-1 and VCAM-I. The sequence identity is 23% over 80 pairs of aligned residues.

As with sequence alignments, structural alignments and superimpositions are not trivial problems and many different answers can be obtained depending on the algorithms that are tried. At this stage of the project I spent some weeks implementing a C++ program for progressive multiple structural alignment, called *msuper*, based on the published work of Russell & Barton (1992) and Gerstein & Levitt (1996). The program turned out to be important for the whole project, whenever structural superimpositions were needed. However, we decided not to use it for a benchmark of alignment quality, since as similar methods, it is found to be unstable in cases where there is no sequence similarity at all. Therefore, we preferred an external program, SSAP (Taylor & Orengo, 1989),

Figure 2.2: Spatial significance of a sequence alignment. The alignment shown in Table 2.2 is used to guide the superimposition of the backbone coordinates of the corresponding PDB structures (1ij9 (Taylor *et al.*, 2001) and 1d3l (Kolatkar *et al.*, 1999)). Residues are coloured according to their chemical properties using Rasmol (Sayle & Milner-White, 1995). Despise the low sequence identity between these proteins, the resulting $C\alpha$ RMSD is 2.46Å.

a well recognised dynamic programming algorithm which uses both sequence and local structure information to superimpose and align three-dimension protein structures. For example, SSAP is used to generate the fold libraries needed by the successful fold recognition program 3D-PSSM(Kelley *et al.*, 2000). An important disadvantage of SSAP is that the superposition file generated by the program contains only the Cα of each residue and cannot be applied to more that two structures at a time. Whenever full atomic details were needed throughout this work, or more than two templates needed to be superimposed, *msuper* was used (see Appendix A for more details).

After several CASP experiments it is generally accepted that, for templates over 50% sequence identity, very similar sequence alignments are obtained regardless of the methods used. However, below this threshold sequence alignments start to diverge from structural superimpositions. Therefore, remote homologous sequences found to have a very similar fold in SCOP, cannot be correctly aligned. To understand this problem and to contribute to solving it, we set up an experiment to test different alignment procedures. The idea was straightforward: take a pair of proteins from every SCOP family and align them by sequence and by structure; then compare both alignments, score its agreement and record their sequence similarity. Several resources were needed:

i Our own dynamic programming implementation (written in C++).

ii Different scoring schemes for sequence and structure matches.

iii A program to compare two alignments and score their agreement.

iv A test set of random pairs of homologous protein domains (taken from SCOP).

Sequence alignments by dynamic programming. Recent work from several groups has enhanced the value of using secondary structure information to improve alignment sensitivity - the ability to recognise remote homologous sequences (see for example (Kelley *et al.*, 2000)). The secondary structure (SS) of a template is easy to assign automatically using popular programs such as DSSP (Kabsch & Sander, 1983) and STRIDE (Frishman & Argos, 1995). For the query sequence, however, prediction programs are needed. Although a number of different algorithms have been used, often these programs are based on neural networks and predict a three-state secondary structure: helical(H) residues, strand(E) residues and coiled(C) residues. Examples are PSI-PRED (Jones, 1999) and PHD (Rost, 1996). The accuracy of these predictors has been established around 70-80% (Rost & Eyrich, 2001). In addition, it has recently been recognised that the use of sequence profiles can increase the sensitivity of alignments, instead of using standard scoring matrices such as BLOSUM or PAM. We decided to incorporate these two concepts into our sequence alignment implementations with the aim of improving alignment accuracy, not necessarily sensitivity. To generate sequence profiles for each sequence PSI-BLAST was used, generating simultaneously the checkpoint file needed by PSI-PRED to do a SS prediction.

Calculating alignment shifts and scoring alignments. To compare our alignments to those generated by SSAP, a perl program was written following a shift scoring function published by Cline (2000), the shift score. This function scores the similarity between two alignments in the range -0.2 (nothing in common) to 1 (identical).

2.2.1 Alignment comparisons

After developing the tools, three conceptually different sequence alignment protocols were tested: pairwise *Clustalw* as a standard global alignment program (with default Gonnet matrix), *Profile1* and *Profile2*. All three methods use Needleman-Wunch-related algorithms. *Profile1* is a global alignment method that uses the a PSI-BLAST-generated PSSM of the query and SS information for both the query (predicted SS_q) and the template (SS_t, as defined by DSSP). Matches across the dynamic programming matrix are scored combining the log-odds of the relevant row of the PSSM and a SS agreement criterion (add +1 if $SS_q = SS_t$). *Profile2* is a method to align the query PSSM to a PSSM of the template, similar to that published by Rychlewski *et al.* (2000), adding the weight of the

SS matching as before. To calculate a match score in the dynamic programming matrix, the dot product of the relevant PSSM rows is taken. A more graphical explanation of these methods is shown in Table 2.3.

Clustalw (Gonnet)	**Profile1**	**Profile2**
sequence to sequence	profile+SS_q to sequence+SS_t	profile+SS_q to profile+SS_t
	HHHCCCCC	HHHHHCCC
	...	...
	VFIWQSSW	AYLFQST-
	AYIWQS--	AYIWQS--
AYLWQSTW	**AYLWQSTW**	**AYLWQSTW**
AYVWQS-Y	**AYVWQS-Y**	**AYVWQS-Y**
		AYLWNSTW
		VYVWNT-F
		...
	HHHHCCCC	HHHHCCCC
232843-2	232832-1	232823-0

Table 2.3: Graphical explanation of the three tested alignment methods, where the query sequence is in the top half and the template in the bottom half. The secondary structure on top is predicted using PSI-PRED. For the template, the secondary structure is parsed from the output of the DSSP program. Query and template are shown in bold. Profile sequences are also shown aligned to them. The last row shows a hypothetical residue bit-score for each column in the alignment. The average of these values along the alignment is defined as the bit-score.

These methods were benchmarked against SSAP using the shift score function. From the initial set of 428 alignments of pairs of SCOP domains, we first had to identify random alignments. Inspecting the distribution of scores we found a strong association between bad shift scores (less than 0.5) and the dynamic programming score divided by the alignment length. This score was called *bit-score*, since it is given in bits. By taking a bit-score cut-off of 2.0, at least 94.5% of the alignments over this value had good shift scores (over 0.5), but at the cost of missing 5% of relatively good alignments. Therefore, the first finding from this experiment was a numeric filter to reject incorrect alignments. The remaining 240 alignments with bit-score $\geqslant 2.0$ are shown in Figure 2.3.

Overall, the *Profile1* method seems to be better, particularly in the lower end of the % sequence identity interval, as shown in Table 2.4 and Figure 2.3. As the distributions of scores are not normal, it is not possible to assess the statistical significance of these

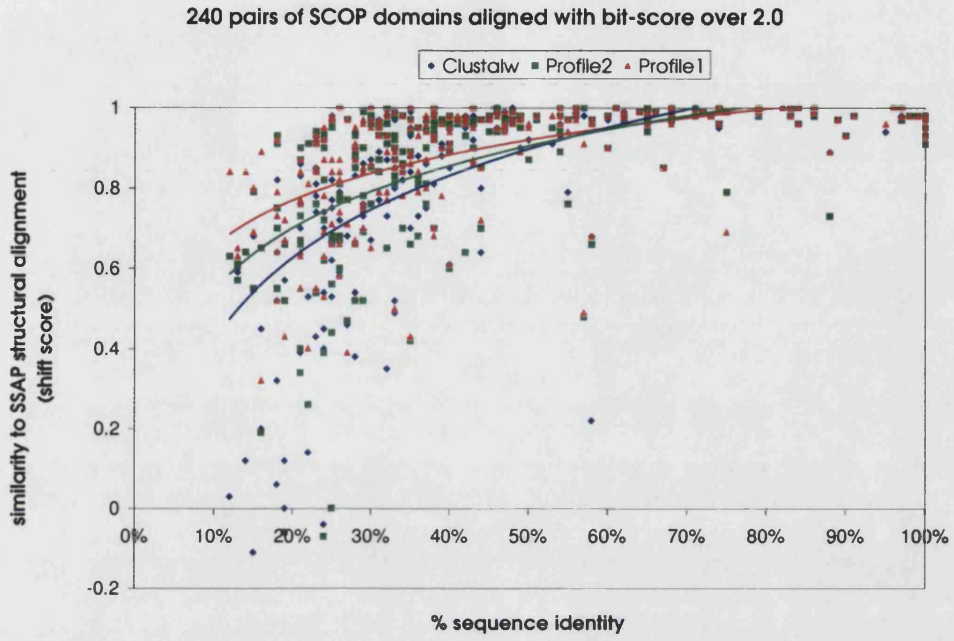


Figure 2.3: Clustalw, Profile1 and Profile2 alignment procedures as compared to SSAP structural alignments. Least-squares logarithmic fitted functions are depicted using the same colour scheme. The correlation between the shift score and % sequence identity is below 40% in all cases. Almost identical results are obtained if *msuper* alignments are used as a reference (see Appendix A)

average differences using a t-test. In addition, the correlation between the shift score and the % sequence identity is quite poor (under 0.4%) for the three methods, allowing no simple rules to be deduced based on % sequence identity to predict alignment errors at the low end of the % sequence identity interval. However, the correlation of the bit-score and the shift score is much better, around 0.7-0.75 for the three methods when fitted to a logarithmic function.

It was observed that in some cases the *Profile1* protocol underperformed if compared to the other two procedures, suggesting, as also reported by Elofsson (2002), that, more generally, no single alignment procedure is consistently better than the others and better alignments would be obtained by probing different methods and being able to identify the best for each particular case. With the data we have, the bit-score is the best estimate we can make as to how good an alignment is in terms of agreement to the reference structural alignment.

	% sequence identity [0-100] (n=240)	% sequence identity [0-35] (n=133)
Clustalw	0.82	0.69
Profile2	0.85	0.73
Profile1	0.88	0.80

Table 2.4: Average shift scores of three sequence alignment procedures as compared to SSAP reference structural alignments.

2.3 Splitting protein domains

After exploring some of the complexities of sequence alignments, and having implemented several alignment routines, we now considered another important problem in modelling: the search for the best templates. For this purpose a good sensitivity is required, to find remote templates, and also accuracy, to get the correct alignments. Furthermore, biological knowledge of the candidate query protein would probably improve the selection of suitable template(s), but this is not easily obtained from a sequence matching procedure such as PSI-BLAST. This situation is even more complex, since proteins may have several domains and therefore different template(s) may be necessary to model each of them. In order to address these problems, once again different procedures were initially considered:

i Construct a library of protein domain families (from SCOP) where each family is represented by a multiple structural alignment and scan it with our own profile alignment procedure, written in C++.

ii Use the PFAM library (Sonnhammer *et al.*, 1998) of protein domain families where each family is represented by a multiple alignment, derived from a manually inspected seed sequence alignment, and scan it using the profile search program IMPALA.

iii Construct our own database merging the PDB and the PFAM sequences and scan it with PSI-BLAST.

The first procedure relied on maintaining a large library of protein multiple structural alignments. Although our alignment routines were ready to test, we didn't favour this approach as this complex library, in a real scenario, would need weekly updates to include new proteins added to the PDB into their corresponding multiple structural alignment. This posed several problems, such as splitting these new PDB entries into their corresponding SCOP domains (precisely the problem we were trying to solve) and generating,

automatically, a large number of structural alignments. This did not seem an easy task. Another important drawback was that this sort of library would not represent the whole protein sequence database, because SCOP and the PDB are only a small fraction of it. Eventually this path was not further explored.

The second procedure seemed easier to benchmark since PFAM is updated frequently and family multiple alignments are already built by their developers. In addition, many families already contain information about which PDB structures are related by homology. Furthermore, PFAM covers many protein families for which no structural information is available, families not represented in SCOP. By using PFAM the tasks of splitting domains and finding templates are separated. Domains can be confidently identified even when suitable modelling templates cannot be found. PFAM A+B were downloaded and 300 random protein sequences, 105 of which sharing less than 30% identity to their respective PFAM families, were extracted. Attempts were made to match each of these test sequences to the right PFAM family out of the total number of 3360 families (PFAM7.0), by using the program IMPALA with default parameters. It must be stated that these 300 sequences were removed from their original PFAM families to make the experiment more realistic. Results are shown in table 2.5:

PFAM library	inclusion of NCB	low-complexity filtering	best hit = correct family
PFAM(A+B)	+	+	290/300
PFAM(A+B)	-	+	290/300
PFAM(A+B)	+	-	293/300
PFAM(A+B)	-	-	293/300

Table 2.5: Performance of IMPALA identifying 300 random PFAM protein families. NCB are non-conserved blocks in a PFAM multiple alignment, usually shown in lower case in their original Stockholm format. When indicated, low-complexity regions were masked during the IMPALA search. In either case, the IMPALA procedure failed to correctly identify the PFAM families for 7 to 10 test sequences.

The third and final procedure consisted of constructing a single sequence database by merging the PDB (in fasta sequence format) and all the sequences extracted from PFAM A+B families, storing in their headers the family they had been extracted from. This database was named dPFAM_PDB. The same 300 test sequences were scanned using PSI-BLAST with default parameters and in this case all of them were correctly assigned to their respective families. The added advantage of this method is that it potentially permits the identification of multiple PFAM domains in a single PSI-BLAST search. Since sequences in the database are labelled according to their respective PFAM families, by

processing the PSI-BLAST output it is possible to read the most probable domain assignments from the N to the C-terminus of the query sequence. Two iterations of PSI-BLAST are enough for this purpose. Another advantage is that PFAM families often contain PDB templates that the PSI-BLAST search alone cannot identify. However, if at least one sequence in the family is confidently aligned to the query, with a good *e*-value, then the template could be matched as well, by collapsing the multiple alignment, as shown in Table 2.6.

2.4 Domain Fishing, a first step in Comparative Modelling

As a way to make these methods, described in the previous Sections, available to the community, we decided to design a web server implementing these tools. The server aims to help the user in the process of selecting and aligning templates for Comparative Modelling tasks. The program is called DomainFishing and has been live on the World Wide Web (http://www.bmm.icnet.uk) since November,2001, completing more than 8000 jobs in its first 20 months.

This server, made public through the journal *Bioinformatics* (Contreras-Moreira & Bates, 2002), is best described in the flow chart in Figure 2.4.

These are the main steps:

1. First the query sequence is scanned against dPFAM_PDB with two iterations of PSI-BLAST, reporting in the output all the hits, to allow identification of every domain in cases where many hits match the same region of the query. For example, if the query protein contains an immunoglobulin domain it will match thousands of sequences in dPFAM_PDB and those could hide remaining domains, by flooding, in the output.

2. Definition of domains. Given that PSI-BLAST hits are ranked according to their e-values, the output is scanned recording non-overlapping hits that maximise the coverage of their PFAM families.

3. Possible functional annotation for each domain is extracted from the relevant PFAM families.

4. PDB templates are extracted from the PSI-BLAST output, and from the domain-defining PFAM families, and mapped to the relevant domains along the query se-

```
Q9N629              TPNHLLTLLI-t---KRKICILEAASGDEaksRDAFSVDHIESARLIF...
Q9Y6W6              MTKCSKSHLP-----SQGPVIIDCRPF------MEYNKSHIQGAVHIN...
QQ9AG15             VTESLVALLE--S-gTEKVLLIDSRPF------VEYNTSHILEAININ...
Q9BSH6              TVAWLNEQLElg---NERLLLMDCRPQ------ELYESSHIESAINVA...
Q91790              LKALLAERAH-------KCLILDCRSF------FSFSSCSIVGSSNVR...
DUS1_RAT            DAGGLRALLRer---AAQCLLLDCRSF------FAFNAGHIVGSVNVR...
Q13524              SHGTLGLPSG------GKCLLLDCRPF------LAHSAGYILGSVNVR...
PYP2_SCHPO          TLKSFEEQTE------SVSWIIDLRLH------SKYAVSHIKNAINVS...
PTP3_YEAST          TAVELGKIIEtlp--DEKVLLLDVRPF------TEHAKSIITNSIHVC...
Q9P080              VTGHFKTPSKktKssKPKLLVVDIRNS------EDFIRGHISGSINIP...
YOUA_CAEEL          IMQKLSQIEF-----MQKYILIDCRYD------YEYNGGHIKGAQSLF...
TWIN_DROME          IQGEFDEQLG-s---QGGYEIIDCRYP------YEFLGGHIRGAKNLY...
MPIP_DROME          LKGEFSDKVA-------SYRIIDCRYP------YEFEGGHIEGAKNLY...
MPI1_XENLA          IHGDFSSLVE-------KIFIIDCRYP------YEYDGGHIKGALNLH...
MPI1_HUMAN(1C25)    LNGKFANLIK-------EFVIIDCRYP------YEYEGGHIKGAVNLH...
Q9IAA8              ----------------------DCRYP------YEYEGGHIKGALNLH...
MPI2_RAT            LTGKFSNIVE-------KFVIVDCRYP------YEYEGGHIKNAVNLP...
UBP4_YEAST          SANSASSQME--------ILLIDIRSR------LEFNKSHIDTKNIIC...
PYP1_SCHPO          LQEYLDKEAW-----KDDTLIIDLRPV------SEFSKSRIKGSVNLS...

query               SCLWLRRELSPPRPRLLLLDCRSRELYESARIGGALSVA...
Q9BSH6              TVAWLNEQLELGNERLLLMDCRPQELYESSHIESAINVA...
                                      |||    ||   |  |    ...
MPI1_HUMAN(1C25)    LNGKFANLIK----EFVIIDCRYPYEYEGGHIKGAVNLH...
```

Table 2.6: Finding and aligning templates within PFAM families. Dual specificity human phosphatase 9 (DUS9_HUMAN) was used to scan dPFAM_PDB. The N-terminus can be annotated, through PFAM, as a Rhodanese-like domain, PFAM family PF00581. A simplified version of this family is shown here on top. Q9BSH6 was confidently matched by PSI-BLAST with an *e*-value of $1e^{-45}$. The PDB template 1C25 (Fauman *et al.*, 1998a), not found by PSI-BLAST, is also highlighted in bold. The second box of the table shows the PSI-BLAST alignment of the query and Q9BSH6. By collapsing the multiple alignment, a pairwise alignment to 1C25 can be obtained, yielding just 16% sequence identity. The quality of these alignments relies on the quality of PFAM multiple alignments.

quence.

5. Since domain definitions in SCOP are related to their spatial structure in experimentally solved structures they are perhaps more useful from a modelling perspective,

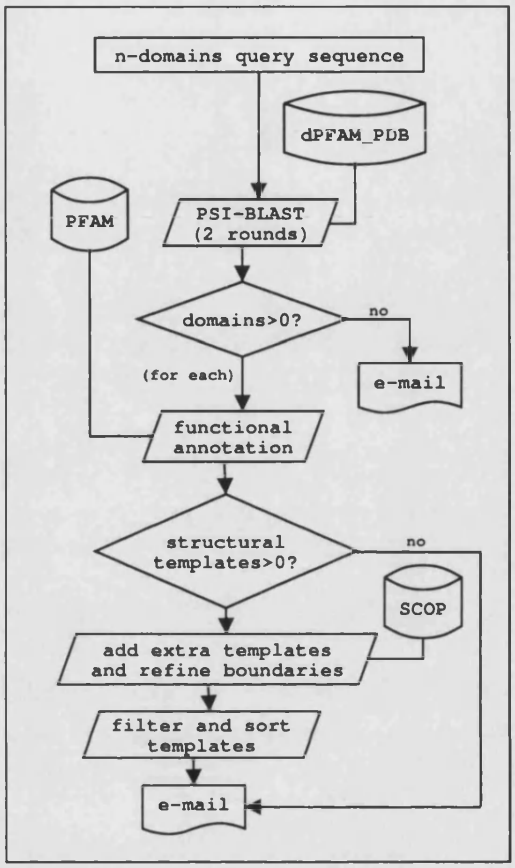


Figure 2.4: Flow chart of Domain Fishing.

so domain definitions from SCOP are used to trim template boundaries. Extra templates, not found by sequence similarity in the initial PSI-BLAST search, may be obtained from the SCOP families containing our templates.

6. The query sequence is split into the PFAM-defined domains and each of them is iteratively aligned to all the available templates. In particular, following the lines suggested in Section 2.2.1, different alignment procedures are used (using template's PSSM, query's PSSM or both, see Section 2.7 for more details), trying to extend the query sequence to maximise the template coverage. Bit-scores are used to discriminate between them. Eventually only one alignment per template is reported. Templates are ranked according to the product of % sequence identity to the query and the coverage of it (*coverID* score). When possible, crystallographic resolution is used to differentiate identical *coverID* scores.

7. Results are sent back to the user using a HTML report. Rasmol-based three-dimensional backbone models corresponding to each alignment are also provided, with residues coloured according to the relative conservation within the domain's protein family. These models can also be useful to detect alignment errors, as shown in Figure 2.5. Details on how this is done are given in Section 2.7. Rasmol was chosen because it is available in virtually all platforms. In addition, a link to use these alignments to build models with 3D-JIGSAW is attached. These alignments can also be edited by the user.

An example of DomainFishing is now shown. Dual specificity human phosphatase 9 (DUS9_HUMAN), taken from the SWISS-PROT database (Boeckmann *et al.*, 2003), was selected to be used as input for DomainFishing. The program identified two domains: a rhodanese-like domain (PFAM family PF00581) for the first 130 residues and a catalytic domain of a dual specificity phosphatase (PF00782) for residues in the interval [200-340]. For the N-terminal domain, the template 1HZM (Farooq *et al.*, 2001), a 44% identical ERK2 domain of MKP-3, is selected on top of the list. For the C-terminal domain, 1MKP (Stewart *et al.*, 1999) is selected with 80% of sequence identity. The alignment of the C-terminal domain to 1HZM is shown in Table 2.7.

The presented tools have also been incorporated into the Comparative Modelling web server of the laboratory, 3D-JIGSAW (Bates & Sternberg, 1999), to find, align and rank templates and define domains. In the interactive mode the server allows the user to select templates, to manually edit alignments and to actually build models. In the automatic mode these steps require no intervention, the process is completely automatic. Since then 3D-JIGSAW has joined the EVA evaluation project (Eyrich *et al.*, 2001), for which some results were shown in Figure 1.9. As of September 2003, the overall performances of the servers participating in EVA are summarized in Table 2.8. In this comparison, 3D-JIGSAW, using these alignment procedures, is shown to be as accurate on average as the other servers, but with the ability to perform equally on difficult cases.

2.5 Conclusions

During this part of the work we observed that, as far as protein Comparative Modelling is concerned, none of our sequence alignment techniques can be considered to be perfect, in agreement with observations in the literature. Although it is possible to rank them, in certain situations 'weaker' techniques can perform better than 'stronger' ones. These facts suggest that several alignment techniques should be used to generate a variety of

Query	residues 1 to 145
% identity	44
bit-score	3.58
%CoverID	42
Resolution	(NMR)
Parameters	1 0 1 0.25 1

```
query       ------------MEGLGRSCLWLRRELSPPRPRLLLLDCRSRELYESARIGGALSVALPA
                        +++++++||+++|+ +++||||+|||+ |||||++|  |++||+|
1hzm_A      MIDTLRPVPFASEMAISKTVAWLNEQLELGNERLLLMDCRPQELYESSHIESAINVAIPG
ACCI        629519585612315313111156515723774111111445486126211211112175
SS_qp       ------------CCCHHHHHHHHHHHHCCCCCCEEEEECCCHHHHHCCCEEEEECCCCHH
SS_tk       CCCCCCCCCCCCCCCCCCCCCCHHHHHHHCCCCCEEECCCCCHHHHHHCCCCCCCCCCC

query       LLLRRLRRGSLSVRALLPGPP-----LQPPPPAPVLLYDQGGGRRRRGEAEAEAEEWEAE
            ++||||++|+|+||||+++++      +++   +|+|||+ +++         ++|++++|
1hzm_A      IMLRRLQKGNLPVRALFTRGEDRDRFTRRCGTDTVVLYDESSSD--------WNENTGGE
ACCI        116541233716441115855145214754835111111112311--------16656983
SS_qp       HHHHHHCCCCCCCCCCCCCHH-----HCCEEEEEEEEEECCCCCCCHHHHHHCCCCCCCH
SS_tk       HHHHCCCCCCCCCCCCCCCCCHHHHHHHHCCCCCCEEECCCCCCC--------CCCCCCCC

query       SVLGTLLQKLREEGYLAYYLQGGFSRFQAECPHLCETSLAGR
            |+||+||  ||++||++|+||+||||+||||++ +|||+|+|+
1hzm_A      SLLGLLLKKLKDEGCRAFYLEGGFSKFQAEFSLHCETNLDGS
ACCI        64126117314877241111252866247324641253255*
SS_qp       HHHHHHHHHCCCCCCCEEEECCCHHHHHHHHHHHHCCCCCCCC
SS_tk       CHHHHHHHHHHHCCCCCEECCCCHHHHHHHHCCCCCCCCCCC
```

Table 2.7: DomainFishing sample alignment of DUS9_HUMAN rhodanese-like domain and its closest template. According to Figure 1.9, a model based on this alignment is predicted to have a RMSD to its experimentally determined structure of less than 2Å on the best case and about 2.8Å on average. Note the differences between the predicted and the template three-state secondary structure. ACCI is the relative solvent accessibility for each residue of the template, SS_qp is the predicted secondary structure of query and SS_tk is the DSSP secondary structure of the template, where H=helix, E=beta-strand and C=coil. The aligmnent parameters are: SS_match, SS_mismatch, gap_opening, gap_extending, PSSM used (template in this case). | and + signs mark identical and similar residues according to the sequence profile(s) used, that is, residue bit-scores.

server	chains	weeks	%cover	%equiv.positions	difficulty
3djigsaw	4496	59	96	88	22
cphmodels	1228	72	96	87	8
esypred	5346	77	96	86	21
sdsc1	2667	48	93	80	17
SwissModel	9316	153	93	87	16

Table 2.8: Performances of Comparative Modelling servers participating in EVA. %cover is the percentage of modelled residues with respect to target length. %equiv.positions is the percentage of equivalent $C\alpha$ positions within 3.5Åbetween the optimally superimposed target and model structure. Difficulty ranges from 0 (easy cases) to 100 (difficult cases). The difficulty level is defined as the percentage of missaligned residues between an optimal structural superimposition alignment and a pairwise sequence alignment method for the modelled region. If the sequence alignment is identical to the structural alignment, the difficulty is 0. If there is no similarity in any structural alignment (using the program CE (Shindyalov & Bourne, 1998)), the difficulty is 100. Note that 3D-JIGSAW uses the DomainFishing algorithm to find, select and align templates.

Figure 2.5: Rasmol-based conservation map of the DUS9_HUMAN rhodanese-like domain modelled with 1C25 (Fauman *et al.*, 1998b), as aligned by DomainFishing. The protein $C\alpha$ trace is coloured according to the conservation of each residue in the stacked multiple alignment calculated by PSI-BLAST, where blue residues are not particularly conserved and red are the most conserved. Asp26,Cys26 and Arg27, conserved catalytic residues in the domain family, are in red. Note that the residues are hotter as they get closer to this functional center of the molecule. This suggests that the alignment quality within this family will be higher close to these residues, and lower in other areas, since there the sequence is much less conserved. The overall bit-score of this alignment is 2.9, despite just 17% sequence identity.

alignments. However, we should then be able to distinguish good from bad alignments. Evaluators such as sequence identity, coverage or *bit-scores* could help. We explore these

issues further in the next chapter. Despite these limitations, the web server DomainFishing was designed and implemented to help the user in the first steps of a Comparative Modelling job: defining domains, finding templates and aligning them. This tool is also linked to the Comparative Modelling server 3D-JIGSAW, so the user can build protein models easily and interactively, with the aid of some quality evaluators such as *bit-scores* or 3D-conservation maps. Both servers are extensively used by the community and their performance can be monitored through the EVA automatic continuous evaluation. 59 weeks of automatic evaluation show that our alignment procedures, together with 3D-JIGSAW methods for actually building models, are among the best servers in terms of coverage and accuracy , with the ability to model more difficult cases.

2.6 Possible developments

The use of residue bit-scores allows mapping along a sequence alignment regions with high or low confidence. This data could then be used to automatically detect weakly aligned parts of the alignment, indicating regions where alternative alignments should be considered. Indeed, recent work explores this possibility (Tress *et al.*, 2003).

We have noticed that DomainFishing usually underperforms in finding and correctly aligning very remote homologous templates when compared to Fold Recognition programs. Improving these tasks would enhance the server as it stands now.

Often DomainFishing reports several short PFAM domains that taken together in space make up a single structural domain defined by SCOP. In these cases it may be a good idea to generate a unique alignment comprising all these PFAM domains.

The current implementation of the program treats templates found through the PSI-BLAST search of dPFAM_PDB and those inherited from PFAM families differently. This produces two separate lists of templates and domain boundaries. It may be better to generate a single list of templates, including alternative alignments for each. However, re-ranking of templates in this list would not be trivial as it may require feedback from the three-dimensional structures and has to consider coverage of one or more domains and alignment accuracy. This issue will be partially addressed in the next chapter.

When several domains in a protein are modelled independently, usually they are in different spatial frames of reference. A possible improvement for this tool would be to automatically derive restraints at the sequence level and use also the available multidomain structure information from templates to calculate the most probable conformations for each individual domain relative to each other.

2.7 Some methodological details

dPFAM_PDB is created by merging PFAM A+B and the amino acid sequences from the PDB structures, particularly the sequence as contained in the ATOM records, the experimental sequence. Low complexity regions are filtered out from these sequences using the program SEG (Wootton & Federhen, 1996), since they may affect PSI-BLAST alignment scores.

The DomainFishing PSI-BLAST search is done with this command:
blastpgp -i query -d dPFAM_PDB -b 100000 -v 100000 -j 2 -s T -C chkfile -Q pssmfile
using NCBI blast version 2.2.5. -b and -v are used to display all the generated alignments, not just the top 1000; -j 2 for two iterations and -s T to filter the query for low complexity. The checkpoint file *chkfile* is kept to predict the three-state secondary structure with PSI-PRED 2.3. The *pssmfile* is used to align domains in the query to templates using the *Profile1* and *Profile2* methods. It is also used, as mentioned in Section 2.4, to extract the information per residue, as an indirect measure of conservation (column 23 in the file). This information is scaled in the range [0-99] and it is stored in the temperature factor column (PDB format) in the primitive three-dimensional models for each alignment. These primitive models are built by inheriting the backbone coordinates of the aligned template.

SSAP sequence alignments were obtained by processing the original output from the program and filling the missing, not aligned residues, as insertions. This script was written in perl.

To transform the original seven-state DSSP secondary structure assignments to three-states, alpha, π and 3_{10} helices are considered just as helices, extended strands are conserved and the remaining states are labelled as coiled.

The global dynamic programming routines *Profile1* and *Profile2* implemented for this work include the criterion first proposed by Gotoh (1982) to speed up the calculation of gaps in equation 2.1. For efficiency purposes, when different alignment procedures are going to be tested on the same pair of protein sequences, as DomainFishing does, the dynamic programming and the trace-back matrices are allocated only once. Then the process iterates through a set of parameters, as mentioned in Figure 2.7, recalculating these matrices each round. Several alignment procedures can be attempted:

When templates were aligned to PFAM-defined domains in the query, efforts were made to extend the alignment. Since templates are trimmed according to SCOP annotation, query domains were extended towards both the N and C termini in order to cover the entire template. In DomainFishing, extended alignments are only preferred if their bit-score is at least 60% of the original, not extended, bit-score.

iteration	$SS_q \neq SS_t$	$pssm_q$	$pssm_t$	comment
1	0	+	-	*Profile1*
2	-1	+	-	
3	0	-	+	
4	-1	-	+	
5	0	+	+	*Profile2*
6	-1	+	+	
7	0	+	+	$\frac{pssm_q+pssm_t}{2}$, as in 3D-PSSM
8	-1	+	+	

Table 2.9: Dynamic programming parameters used in this work and in DomainFishing. Gap opening and extending costs were kept constant, 1 and 0.25 respectively, and secondary structure matches were scored with +1 (other values were tested with no apparent difference). Procedures that require the pssm for the template ($pssm_t$) are only performed if those profiles are kept in a library, otherwise they will require PSI-BLAST to be run for every template. The 3D-PSSM way of combining information from two PSSMs was not extensively benchmarked in this work, since that is published work (Kelley *et al.*, 2000) and the performance of that method was outstanding, for example, in CASP4 (Bates *et al.*, 2001).

Chapter 3

Recombination of protein models

In the next three Sections some experiments that led us to test a new approach in Comparative Modelling are described. In particular, we concentrated on the first three steps of the generic comparative modelling procedure, as shown in Figure 1.7: template selection, query to template alignment and single/multiple template modelling. For this, the program 3D-JIGSAW was used, which has been shown to be competitive in previous CASP experiments (Bates & Sternberg, 1999; Bates *et al.*, 2001) and in EVA (see Figure 1.9). However, we do not consider that the results presented here are significantly sensitive to the choice of a particular CM program. The remaining Sections are dedicated to the presentation and benchmarking of the new approach, termed *in silico* Protein Recombination. Unless otherwise stated, in the following experiments a minimum difference of 0.6Å was used to compare RMSD measures. As mentioned in Sections 1.2.1 and 1.2.2, this value has been found to be the average backbone variability observed between protein structures solved under different crystal lattices or when comparing NMR and crystallographic structures.

3.1 Sorting templates

In Sections 1.4 and 2.6 some of the difficulties of correctly sorting Comparative Modelling templates were introduced. A more thorough investigation of this is now presented here, since this is still considered, at least within the CASP community, to be a major problem affecting the quality of comparative models (Tramontano *et al.*, 2001).

Chothia & Lesk (1986) quantified the principle of "similar protein sequences have similar folds" on a small number of pairs of proteins. For each pair, they defined the protein core as the fraction of residues that can be superimposed within 3Å of RMSD,

using $C\alpha$ coordinates. Finally, they proposed a function to relate sequence identity and structural similarity, Equation 3.1, by least-squares fitting the data they had (see also Equation 3.5).

$$RMSD_{expected} = 0.40e^{1.87(\frac{100-\%sequence\_identity}{100})} \tag{3.1}$$

The data shown in Figure 1.9 would fit to a similar function. Thus, it seemed reasonable to rank the possible templates to build a model using their sequence identity to the query. Indeed, one of the most successful programs for comparative modelling, Swiss-Model(Guex *et al.*, 1999), weights the contribution of each template to the final model using exactly this criterion. An experiment was set to further test the validity of this approach. Using 3D-JIGSAW, models for 392 SCOP domains were built using up to four different templates. Each quartet of models was then compared to the experimental structure, see Figure 3.1. This trivial experiment allowed us to estimate the difficulty of selecting templates. Within this dataset, errors in choosing the optimal template are equally likely for each of the sequence identity ranges used, with a frequency of approximately 25%. If structural alignments are used instead of 3D-JIGSAW sequence alignments, sequence identity is indeed a good template classifier, suggesting that alignment errors mask the identification of the best template.

Similar difficulties are encountered if templates are ranked by using PSI-BLAST *e*-values, based on similarity scores, as shown in Section 3.6.3. Being unable to routinely identify the optimal CM templates suggest that using several templates might be necessary. This will be discussed in Section 3.3.

3.2 Optimally aligning the templates

As seen in the previous chapter, the main information types usually available to calculate alignments are protein sequence and secondary structure, and the most used algorithmic approach is dynamic programming. In this Section we analyse:

i How often optimal *Profile1* sequence alignments between query and template, with parameters shown in the first row of Table 2.9, correspond to three-dimensional models with minimum RMSDs to their experimental structure. In other words, how often our best sequence alignments correspond to minimum RMSD models.

ii How often alternative trace-backs, suboptimal in a dynamic programming context, yield better models. In other words, how important is alignment variability for

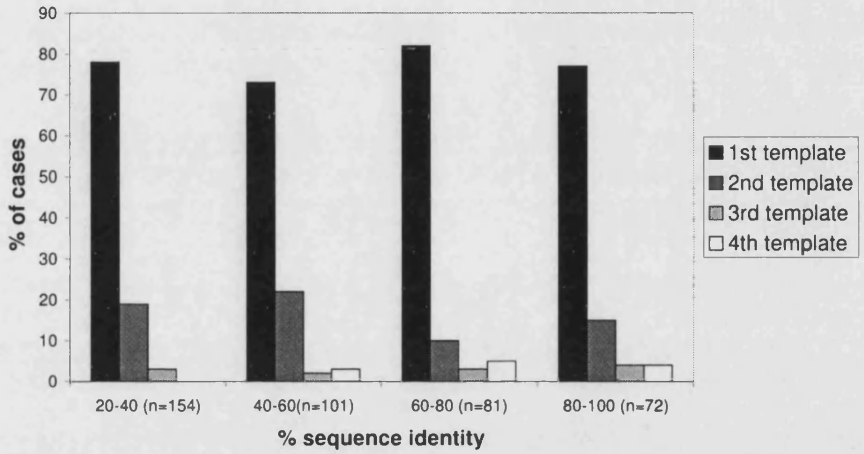


Figure 3.1: Selecting CM templates by sequence identity. For each of the four sequence identity bins, up to 4 potential templates to build a model are ranked according to their %sequence identity to the query sequence. A model is constructed from each and then compared to the experimental structure. The bars represent how often the first, the second, the third or the fourth template yield the best model in terms of RMSD.

Comparative Modelling accuracy. For this we used the procedure published by Saqi *et al.* (1992), explained in Section 3.11.

58 random SCOP domains were picked for this analysis. Using a simple procedure explained in Section 3.11, one optimal and four suboptimal alignments (numbered from 1 to 5) were produced for each of the 58 templates, which had sequence identities in the range [15-82]. For each of these alignments a model was constructed using 3D-JIGSAW and then compared to its corresponding experimental structure. Results are shown in Table 3.1.

Table 3.1: Alternative alignments and RMSD of subsequent models for 58 SCOP domains. Alignment methods are reported in the left column, where 1 stands for the optimal *Profile1* alignment and 2,3,4,5 correspond to subsequent suboptimal alignments, increasingly different from 1. RMSD differences between models are of at least 0.2Å. If a RMSD cutoff of 0.6Å is used, still 10 out 58 cases are better modelled using a suboptimal alignment.

alignment	$C\alpha$ RMSD	SCOP domain	% sequence identity
1	2.68	d1eema2	12
3	2.41	d1hqoa2	15
2	4.51	d1f2ea2	17
2	5.84	d1vcba_	17
1	3.26	d1qqta1	18
1	3.97	d1ndda_	18
1	4.47	d1fb3a1	18
1	4.78	d1gawa1	18
1	6.72	d2phla2	18
4	3.29	d1ubi__	18
1	2.69	d1axda2	19
1	2.72	d1a0fa2	19
1	2.89	d1ljra2	19
1	4.80	d1ndh_1	19
3	3.23	d1qfza1	19
4	3.49	d1c3ta_	19
1	2.10	d1gnwa2	20
1	4.13	d1bfd_2	20
1	4.67	d1fnc_1	20
2	4.31	d1que_1	20
2	4.65	d2cnd_1	20
1	3.15	d1aw9_2	21
1	5.13	d2caua2	21
1	5.63	d1psra_	21
1	2.36	d1gsea2	22
1	3.13	d1pmt_2	22
2	2.56	d1duga2	22
3	3.10	d1poxa2	22
1	2.19	d1f3ba2	23

(continued on next page)

Table 3.1:

(continued from previous page)

alignment	$C\alpha$ RMSD	SCOP domain	% sequence identity
1	3.50	d1bt0a_	23
2	3.42	d1fdr_1	23
1	2.44	d1gula2	24
1	2.72	d1fhe_2	24
1	4.02	d1a8p_1	24
1	5.53	d1acf__	24
1	4.61	d1qlsa_	25
2	2.71	d1pd212	26
1	2.42	d1b48a2	27
2	4.20	d1mr8a_	27
1	2.51	d2fhea2	28
1	5.58	d1f2ka_	28
1	5.30	d1a4pa_	29
1	2.76	d3gtub2	30
1	5.49	d1ypra_	30
2	2.60	d1hna_2	30
2	2.72	d2gsta2	30
1	3.40	d1gsua2	32
1	4.61	d1cqa__	34
1	4.90	d1g5ua_	34
1	2.49	d2gsq_2	35
1	4.65	d1e8aa_	35
1	2.96	d1zpda2	38
1	4.24	d3nul__	40
1	5.17	d1mho__	41
1	3.20	d1dgwa_	49
3	3.59	d1euvb_	51
1	2.07	d2gsra2	81
1	1.15	d1glqa2	82

In 42 cases the highest sequence identity alignment provided the lowest RMSD model, but the remaining 16 cases would have been more accurately modelled using a suboptimal alignment. These suboptimal alignments have a range of sequence identities to their templates from 15% to 51%, considered to be the most problematic for alignments (see Section 2.2). These results suggest that suboptimal alignments (and perhaps other alternative alignments) should be routinely considered in model construction rather than relying on the optimal dynamic programming sequence alignment. Indeed, Comparative Modelling servers such as EsyPred3D (Lambert *et al.*, 2002) try to improve its performance by considering alternative and consensus alignments. Of course this raises the question of how to identify the best alignment. At the sequence level the bit-score could be used, but it seems preferable to have a three-dimensional criterion, allowing models obtained from unknown alignments, for instance from web servers not returning alignments or even *ab initio* models, to be compared.

3.3 Comparative Modelling: one or more templates?

In theory, building comparative models from more than one template could improve their accuracy since more conformational space for the backbone can be sampled. It could actually be the key to calculate better protein models than any of the templates used. However, analysis of CASP4 results showed that only very occasionally were multi-template models more accurate than single-template ones. The reasons for this are the choice of templates (reviewed in Section 3.1) and sequence alignment errors (Tramontano *et al.*, 2001; Venclovas, 2001). As the limited number of targets for comparative modelling in CASP4 precluded definitive conclusions, we performed a more exhaustive but simple experiment using 3D-JIGSAW:

i From each of 271 SCOP families, one protein domain (the query) was randomly selected to be modelled, the remainder were used as potential templates. Two different models were constructed, one using the template with the highest sequence identity to the query, as aligned with *Profile1*, and the other using up to 5 templates. In order to minimise alignment errors, each query was aligned to its respective template(s) on the basis of their known atomic coordinates.

ii Both models were compared to the experimental structure and scored according to RMSD.

The results presented in Figure 3.2 show that 3D-JIGSAW single-template comparative models tend to be more accurate than those built several templates. It can be con-

cluded that our current methodology is not taking full advantage of the possibility of using several templates to build comparative models. In general, multiple-template models are no better than their corresponding ideal single-template models and, indeed, can be considerably worse. Only in a marginal proportion of cases using more than one template was found to be an advantage (improving in the best case 1.66Å), but showing no preference for any region in the sequence identity range. On the other hand, multiple-template models could be significantly worse (1.92Å in the worst case) with a comparable frequency. Because these results are similar to those obtained in CASP4 for all the participant methodologies, it is tempting to think this is actually a limitation of the generic CM method itself. In other words, single-template models appear on average more accurate provided that the optimal template can be identified. Errors in the template(s) alignment to the query may be disregarded as the reason for this, because the models in this experiment had been built from structural alignments using the program *msuper*, introduced in Section 2.2 and described in Appendix A.

3.4 The Evolutionary Analogy

So far we have learnt how difficult is to select templates and to get the right alignment. For these reasons it does not seem reasonable to build models from a single alignment or template, but instead combining different alignments and templates could be desirable. We also know that using multiple templates in the same way as 3D-JIGSAW does not help, and unfortunately CASP4 suggested this to be a generic problem, affecting other modelling procedures. Therefore a different combination tool was needed to explore sequence alignment space and the different conformations adopted by different templates.

This problem can be described as a combinatorial optimisation problem, a field of study in which many different algorithms have been applied, among them genetic algorithms (Michalewicz, 1996). Genetic algorithms have recently been used for several applications such as protein folding, protein docking and alignment optimisation (Unger & Moult, 1993; May & Johnson, 1994, 1995; Pedersen & Moult, 1995; Morris *et al.*, 1996; Rabow & Scheraga, 1996; Xia & Levitt, 2002; Petersen & Taylor, 2003; John & Sali, 2003). These algorithms mimic the natural mechanisms of chromosomal mutation and recombination, used throughout evolution to generate diversity in populations. In a biological context, a mutation is a spontaneous change in a nucleic acid base that gets fixed in a tissue, organism or population via the usual DNA replication machinery. Recombination is graphically explained in Figure 3.3, and consists of a crossover between, usually, two homologous DNA strands.

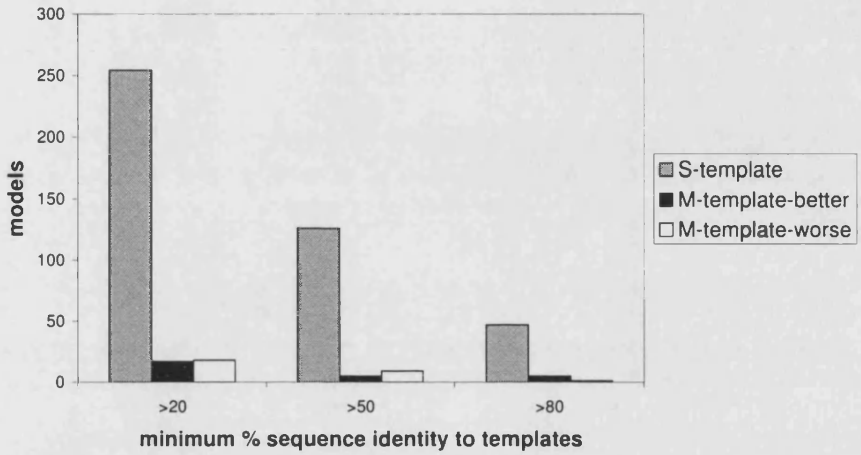


Figure 3.2: Comparing single and multiple-template Comparative Modelling using 3D-JIGSAW and structural query to template(s) alignments. Models were built using 1 or up to 5 templates from the same SCOP family. The bars show how often single (S) or multiple-template (M) models had relatively better or worse RMSD values. The data is split in three % sequence identity bins.

These concepts were taken originally by Holland (1975) to be applied to algorithms. Basically, to apply a genetic algorithm one needs to represent or code solutions for a given problem in a string, just like a DNA molecule. Once this step is done in a satisfactory manner, populations of solutions must be created and then the genetic operators of recombination, mutation and finally selection can do their work. In biological systems, selection tends to favour fit individuals, those who produce more successful siblings that carry at least part of their own genes. In algorithmic terms, fitness usually means how well a solution solves a problem or satisfies some objective function. These concepts are further illustrated in Table 3.2.

In our problem, the idea is to use several templates and different alignments to build a comparative model, expecting to get an optimised final conformation. How are potential solutions encoded? The simplest representation for solutions in genetic algorithms is that shown in Table 3.2, binary strings. These strings can be split into smaller fragments or

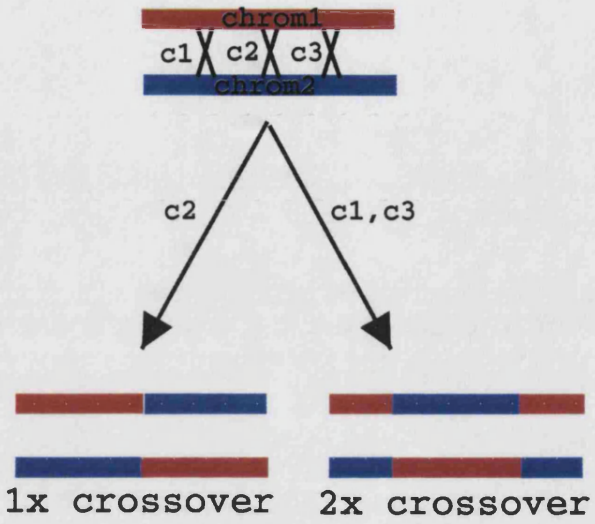


Figure 3.3: Genetic recombination or DNA crossover. Recombination events between chromosomes 1 and 2 with single (c1) and double (c2,c3) crossover points are shown. Double or higher order events are less frequent in real chromosomes.

operator	parameter	$population_{t-1}$ (fitness)	$population_t$ (fitness)
crossover	recombination rate	**10101100** (4) 01010010 (3)	**101011**.10 (5) 010100.**00** (2)
mutation	mutation rate	**10101100** (4) 01010010 (3)	**1010**0**100** (3) 01010010 (3)
selection	selection rate	**10101100** (4) 01010010 (3)	**10101100** (4)

Table 3.2: Basic concepts in genetic algorithms. Potential solutions (chromosomes) inside a population of size 2 are coded here as binary strings. The fitness function in this simple example corresponds to the number of bits with value 1 in each string. A crossover point is marked with a period.

genes, each of them responsible for a given property. In our case, proteins can be seen as implicitly coded solutions, where each residue is a gene and the whole chain is a chromosome built by connecting residues with peptide bonds. Therefore, in our context, models are already encoded solutions, obtained from one template and one sequence alignment:

$$potential\_solution_i = comparative\_model_i = f(template_j, alignment_k) \qquad (3.2)$$

The fitness of a model would then be the likelihood of its fold calculated in an objective manner. Since individual solutions are in this case possible conformations for the three-dimensional structure of the same protein, they have identical sequences and therefore recombination will still be homologous.

3.5 Implementation of the genetic algorithm: *in silico* protein recombination

Taking together these ideas, a genetic algorithm for Comparative Modelling was designed, named *in silico* protein recombination (*insilico*PR) (Contreras-Moreira *et al.*, 2003a). Related combinatorial methods have been applied in laboratory experiments to generate new, viable and useful protein folds, via protein fragment shuffling. This supports this kind of approach (Riechmann & Winter, 2000; Broo *et al.*, 2002).

3.5.1 The method

This method is a genetic algorithm and therefore works at the population level. The input is here defined as a population of atomic detail three-dimensional models for the same amino acid sequence, obtained by Comparative Modelling techniques (plus any other protein modelling methods). The output is another population of models that has survived several generations of artificial selection based on fitness. Recombinant models are derived from the original ones through recombination and mutation. The idea behind this is that the method should be able to conserve good parts from models, combine them in a linear way and discard the rest. In theory, the method should be capable of correcting alignment errors by recombining partial solutions if they are present in the population. Mutation is used to generate novel molecular conformations. The algorithm is outlined in Figure 3.4.

The key steps in this protocol are now described in more detail.

Initial population of models. This population of $1 < size_{ini} \leq 50$ is composed of models obtained from different templates and/or alignments, and potentially from different programs and sources. They must be models for the same primary sequence.

Growing the population. Recombination and mutation. The initial population is grown by randomly selecting pairs of models and applying one of two possible operators:

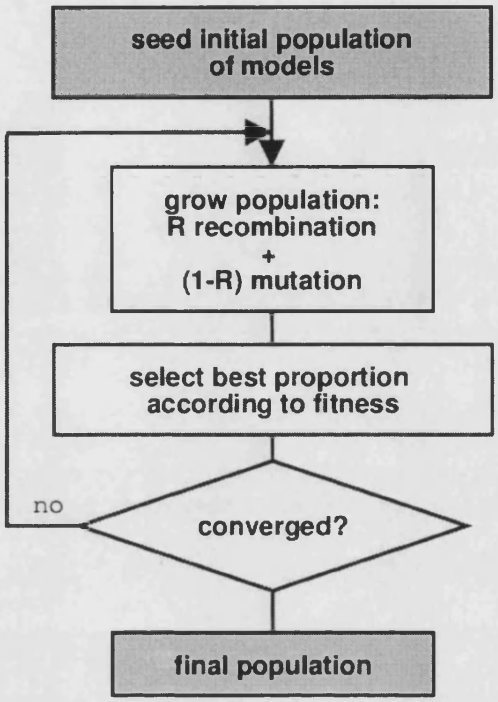


Figure 3.4: In silico protein recombination flowchart. R and $1-R$ are probabilities.

recombination and mutation. In this implementation, these operators are complementary. Every time a pair of models is randomly chosen, they will undergo either recombination or mutation. Recombination occurs with probability R, whilst mutation happens on the remaining $1-R$ cases. Usually values of $R = 0.9$ were used.

In case of recombination, a pair of models is superimposed based on their trivial sequence alignment. This comprises three steps and is also explained in Table 3.3:

i Superimpose them on their $C\beta$ atoms along their entire sequence. Here we use the *msuper* superimposition code, that uses $C\beta$ atoms (see Appendix A).

ii Refinement based only on equivalent residues, those pairs whose $C\beta$ atoms are close in space after the previous step. The tolerance is arbitrarily set to twice the ideal $C\alpha$-$C\beta$ distance (3.61Å).

iii The crossover point is randomly sampled from the set of equivalent residues. In this first implementation, only the subset of equivalent residues not forming regular

secondary structure elements was considered (as defined by the program STICK (Taylor, 2001), see Section 3.11). Only one recombinant model is generated, which inherits the N-terminus from one parent and the C-terminus from the other. STICK is used because it does not compute hydrogen bonds to assign secondary structure and may therefore be more robust to the conformational changes introduced in the process than DSSP.

```
PARENT1   GIFFSTSTGNTTEVADFIGKTLGAKADAPIDVDDVTDPQALKDYDLLFLGAPTWNTG
PARENT2   GIFFSTSTGNTTEVADFIGKTLGAKADAPIDVDDVTDPQALKDYDLLFLGAPTWNTG
SS1       EEEECCCCCCHHHHHHCCCCCCCCCCCCCCEEECCCCCCHHHHHHCCCEEEECCCCCC
SS2       EEEECCCCCCHHHHHHHHHHHCCCCCCEEEEHHHCCCCHHHHCCCCEEEECCEECCC
DISTCB    000011242000111000000012120000000001344454443200000000000
CROSSP    ----xxx-x-----------xxxxxx-------xxx-------xx-----x--xxx

RECOMB    GIFFSTSTGNTTEVADFIGKTLGAKADAPIDVDDVTDPQALKDYDLLFLGAPTWNTG
SSR       EEEECCCCCHHHHHHCCCCCCCCCCCCEEEEHHHCCCCHHHHCCCCEEEECCEECCC
```

Table 3.3: Mechanism of recombination of two comparative models. The sequence alignment bewteen parent1 and parent2, with the distance between $C\beta$ atoms after refining the initial global fitting, is shown. The 3-state secondary structure as assigned by STICK is also shown. The CROSSP row shows the set of potential crossover points (x) in this example, residues that are defined as coil in both models and are less than 3.61Å away. Only one of those points, residue r_i, will be randomly selected, and a recombinant protein made of the N-terminus of parent1, up to r_{i-1}, and the C-terminus of parent2, starting from r_i, is generated. A possible recombinant protein, using the underlined crossover point, is shown in the last two rows. Note that models need not have the same length.

A mutation event requires as well a pair of proteins, randomly chosen, parent1 and parent2. The operation comprises three steps. The first two are the same as in recombination, in order to put the proteins in the same frame of reference and define which residues are equivalent based on their $C\beta - C\beta$ distances (see Table 3.3). The third consists, in this first implementation, of simply averaging the Cartesian coordinates of the equivalent residues. For residues that cannot be paired, because of their $C\beta - C\beta$ distance, the coordinates of parent1 are taken. The idea is to create new conformations in-between the conformations of the chosen parents, something that recombination cannot do. It is a conservative mutation. Of course this would only work with very similar parents, whose atomic coordinates are very close after superimposed, otherwise it would be necessary to rebuild the backbone geometry and the side-chains. This reconstruction code was not added at the time this mutation operator was implemented for the first time, so we knew this was a very limited mutation mechanism.

For both recombination and mutation it is necessary to select randomly two mating protein models. Due to the relatively small populations used in these experiments, mainly due to computing time limitations, we decided to try the following scheme. Initially, all

members p of the population P have the same probability of being picked for mating:

$$prob(p) = \frac{1}{size(P)} \tag{3.3}$$

On subsequent generations, these probabilities are proportionally weighted according to the number of siblings $pastSibs(p)$ that each member p has had in past generations:

$$prob(p) = \frac{1 + pastSibs(p)}{size(P) + totalSibs(P)} \tag{3.4}$$

where $totalSibs(P)$ is the cumulative number of generated siblings since evolution started in this population.

Mutant or recombinant siblings are scanned for bad peptide bond geometries or breaks in the backbone and discarded if the current population requirements are not met. These requirements are updated dynamically after each generation in order to force the population to grow in a limited number (max_{reprod}) of trials. Initially, no more than one main-chain break or 4% of non-planar peptide bonds are allowed; if max_{reprod} is reached, these geometry restraints are relaxed.

Using these operators the population grows until the selection size (usually $2 \times size_{ini} \leq size_{sel} \leq 5 \times size_{ini}$ in our experiments) is reached. At this point the fitness is estimated for every member of the population. The next step is selection. The fitness functions tested are presented in Section 3.5.2.

Selection step. According to the selection rate, only a given proportion (typically 75% in these experiments) of protein models is selected to seed future generations. Smaller selection rates were tried but this one was chosen to avoid potential quality models being lost prematurely, such as models with small sterical clashes after a recombination event, with good backbone geometries. This rate gives them the chance to improve their fitness. In our implementation, therefore, selection consists simply of taking the top 75% members of the population as founders to the next generation.

More sophisticated schemes could be used to reject or accept protein conformations, for example the Metropolis criterion for Monte Carlo algorithms (Leach, 2001). According to this principle, to estimate the difference in fitness between two members of P the temperature of the system should be considered, since small random thermal variations can be allowed. As will be seen in Section 3.5.2, the fitness function used in this work is perhaps too coarse for this and indeed, inclusion of the Metropolis criterion made no difference when tested on a few examples.

Convergence. When the population has converged to similar energies, there is no room for further generation of variability and the evolution process stops. The criterion used to define convergence was initially a fixed fitness (energy) cutoff (0.1 $kcal\ mol^{-1} residue^{-1}$). However, it was observed that this cutoff should depend on the range of energies of the initial population P_{ini}, and therefore the cutoff was set to 10% of the difference in fitness between the best and the worst founders.

This method, implemented in C++ and running under Linux, requires more computing time than more traditional Comparative Modelling methodologies. On a 800MHz Pentium III PC, these simulations can take from a few minutes to several hours, depending on the size and composition of the initial population of models. The algorithm can easily be parallelised although this has not yet been implemented.

3.5.2 Fitness and potential energy functions

Although choosing reasonable recombination,mutation and selection rates is important, the algorithm is critically dependent on the quality of the fitness function. This is, after all, the objective function that the algorithm seeks to optimise, guiding each evolution experiment. In Section 3.4 we referred to this function as the thermodynamical likelihood of a protein conformation. This sort of functions, introduced in Section 1.3.4, can be used to evaluate protein models, since a good model should agree with parameters stored in established force fields. In addition, it is sometimes useful to define these functions as tools to compare model conformations and realistic, experimentally measured conformations. No force fields are needed for this purpose. These fitness functions are useful for benchmarking. RMSD, mentioned in Section 1.2.2, is used as an ideal fitness function in this work to compare evolving models to experimental PDB structures. To calculate the RMSD between proteins p and q they must be first optimally superimposed, to define a set of pairs of equivalent points or atoms in Cartesian space. On this set of size n_{eq} points, usually $C\beta$ atoms in this work, RMSD is calculated as follows:

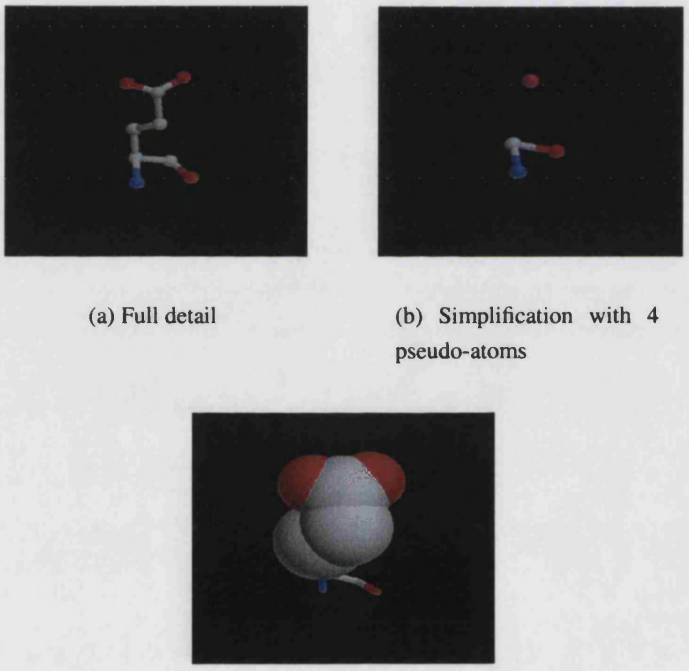
$$RMSD(p,q) = \sqrt{\frac{\sum_{i=1}^{n_{eq}} d_i^2}{n_{eq}}} \tag{3.5}$$

where d_i is the distance between the coordinates of atom i in p and q. This ideal fitness function was used in Section 3.6.1.

When no experimental structural information is known for a protein, or is ignored for a benchmark experiment, potential energy functions can be used to evaluate three-dimensional models. Many sets of parameters and functions have been used over the

years (Robson & Osguthorpe, 1979; Sippl, 1990; Jones *et al.*, 1992; Koehl & Levitt, 1999; Leach, 2001; Russ & Ranganathan, 2002; Wallner & Elofsson, 2003) and so it seemed reasonable to use them as a starting point. The idea was to obtain a simple function, easy to manipulate and to calculate, appropriate for the expected accuracy of the genetic algorithm.

Following work by Levitt (1976) and Robson & Osguthorpe (1979), we tested a simplified representation of proteins, in which residues are made of three backbone pseudo-atoms (C',N',Ca' obtained from the CO,NH, $C\alpha$ groups) plus a side-chain centroid, as shown in Figure 3.5.



(a) Full detail

(b) Simplification with 4 pseudo-atoms

(c) van der Waals volume

Figure 3.5: Simplified representation of residues for our simple fitness function. Here a Glutamic acid is shown in (a). In (b) the same residue is simplified, made of just 4 pseudo atoms. The pink atom occupies the geometric centroid of the side chain and captures part of its chemical properties as tabulated by Robson & Osguthorpe (1979). The C routines and data structures used in this work to implement this simplified representations were coded and provided by Paul W.Fitjohn. In (c) the van der Waals volume for the side-chain is depicted to illustrate how the program NACCESS calculates accessible areas.

Packing of a protein conformation p can then be scored using the atom-atom potentials

derived by Robson & Osguthorpe (1979), using the following Lennard-Jones Equation:

$$potential(p) = interatomic\_contacts(p) = \sum_{i=1}^{n_{atoms}} \sum_{j=i+1}^{n_{atoms}} (\frac{A_{ij}}{r_{ij}^9}) - (\frac{B_{ij}}{r_{ij}^6}) \quad \text{(in kcal/mol)} \quad (3.6)$$

where i, j are pairs of pseudo-atoms in p, A and B are statistical values dependent of the nature of i, j and $_{ij}$ is the distance between them. Note that these potentials implicitly include van der Waals, electrostatic and hydrogen bonding.

Preliminary tests showed that this function was not able to correlate energies and RMSD of protein models consistently. Since proteins fold in solution and explicit solvation terms had been shown to be useful (Holm & Sander, 1992; Koehl & Levitt, 1999), the following explicit solvation contribution was considered:

$$solvation(p) = \sum_{i=1}^{n_{res}} (exposed\_area_i \cdot \Delta Gsolv_i) \quad \text{(in kcal/mol)} \quad (3.7)$$

where $exposed\_area_i$ is the side-chain solvent accessible area of residue i (as calculated using the program NACCESS(Hubbard & Thornton, 1993)) and $\Delta Gsolv_i$ are amino acid solvation free energies tabulated by Eisenberg & McLachlan (1986). NACCESS calculates the area around the van der Waals volume of the side-chain that can be accessed by a water molecule in the context of the rest of the protein. The van der Waals volume for an exposed Glutamic acid is shown in Figure 3.5.

Adding the two terms 3.6 and 3.7 the fitness for protein p can be estimated as:

$$fitness(p) = interatomic\_contacts(p) + solvation(p) \quad \text{(in kcal/mol)} \quad (3.8)$$

As an initial test to evaluate how efficient this fitness function is, it was applied to the 58 models in Table 3.1 to identify the best alignment among the five. It correctly identified the models with lowest RMSDs in 51 cases. Further investigation was carried out to weight the two terms, exploring linear combinations of the terms and their quadratic forms, but eventually a 1:1 weighting seemed to be at least as good as any other linear combinations. More comprehensive tests were subsequently performed (see Section 3.6.2).

3.6 Benchmark of the method

To show how useful this genetic algorithm might be it was necessary to test it first using an ideal fitness function. Only after this could our simple potential energy function be tested on the same data set.

3.6.1 Ideal fitness function: limits of the method

As explained in Section 3.5.2, in our context RMSD is an ideal fitness function (see Equation 3.5). The next experiment was set up to assess *in silico* protein recombination when modelling proteins for which experimentally determined structures are available from the PDB. 163 SCOP domains (32α, 44β, 44 α/β and 45 $\alpha+\beta$ folds) were modelled using their family relatives as templates with the program 3D-JIGSAW. This time *Profile1* alignments were used for these models, one per template. Using the protocol explained in Section 3.5.1 models for the same query sequence were recombined. Results (Table 3.4) show that using several templates in this way permits building models that are on average not significantly more accurate than the optimal template (improvement of 0.46Å), but never worse. However, in some cases the improvement is significant (up to 2.33Å), mainly because of loop choices. For models with no templates over 40% of sequence identity the average improvement becomes significant (0.88Å). From a population point of view, models in the last generation show a consistent improvement (2.6Å better than the initial population). A second important conclusion of this experiment was that mutation does not contribute significantly to the gain in accuracy, as noticed in similar genetic algorithm approaches (Xia & Levitt, 2002). Because we use RMSD as a fitness function, this experiment shows that our algorithm could not further improve regardless of the fitness function applied.

3.6.2 Testing our simple fitness function

Now our energy function was used instead of RMSD. The observed differences in performance can therefore be attributed to the fitness function. We show several sets of results that illustrate the potential of the method.

Correction of alignment errors

First it was decided to check if the method is indeed able to recover alignment errors, as could be expected, since recombination could combine well aligned fragments to built an overall better fragment ensemble. For this, the next experiment was set up. Eight SCOP domains were selected: two α (d1a03a_ and d1a8h_1; shortened to A1 and A2), two β (d1qfja1 and d2phla1; B1 and B2), two α/β (d1pmt_2 and d1poxa2; C1 and C2) and two $\alpha+\beta$ (d1pne_ and d1a5r_; D1 and D2) folds. Models were built for each of them using their experimental PDB structures as templates, but shifting one sequence patch of variable length one, two, three or four positions with respect to its correct place

case	$\Delta RMSD_{population}$ (Å)	$\Delta RMSD_{best\ template}$ (Å)	generations
	Up to 100% identity (N=163)		
Best	-7.49 (-7.60)	-2.33 (-1.77)	1 (3)
Mean	-2.60 (-2.53)	-0.46 (-0.39)*	8 (8)
Worst	-0.16 (-0.23)*	-0.04 (0)*	15 (14)
	Up to 40% identity (N=50)		
Best	-7.49 (-7.60)	-2.33 (-1.77)	2 (4)
Mean	-2.77 (-2.67)	-0.88 (-0.78)	10 (9)
Worst	-0.48 (-0.3)*	-0.05 (-0.01)*	17 (18)

Table 3.4: Benchmark of *in silico* protein recombination using RMSD to the experimental structure as the fitness function. Top: models using templates of any sequence identity. Bottom: only templates below 40% sequence identity had been used. Values in brackets correspond to simulations using only recombination, otherwise mutation has been also applied. The first column shows the final average population RMSD with respect to the founder population average RMSD values. The second column shows the evolution of RMSD with respect to the optimal template, had we initially identified it. Non-significant differences are marked with *. The last column shows how many generations were needed to reach convergence. Significance here refers to RMSD differences smaller than usually observed between proteins solved under different crystal lattices or by NMR.

in the sequence alignment. Thus every initial modelling population was composed of five partially wrong protein models and was fed into the recombination program. Since the genetic algorithm is not deterministic, five replicates for each of the eight SCOP sets were performed. Figure 3.6 shows that this algorithm is able to recombine models to yield better-aligned models, suggesting that it is robust enough to overcome alignment errors if partially correct alignments are present in the initial population. Again this reinforces the view that using models constructed from different alignments should result in more favourable protein conformations.

A more detailed analysis of this experiment, illustrating a typical protein recombination simulation, is shown in Figure 3.7, taking d1pne_ (1PNE,(Cedergren-Zeppezauer *et al.*, 1994)) as an example. In this instance, after generating an initial population in which every member had serious alignment errors, a recombination experiment spanning over 13 generations converged onto a final population in which members had perfect alignments, with RMSD values to the experimental structure of 0.8Å (0.05Å for the backbone). Crossover points found in the final models are shown in the multiple structural alignment of the initial models (A) and in a three-dimensional molecular representation (B).

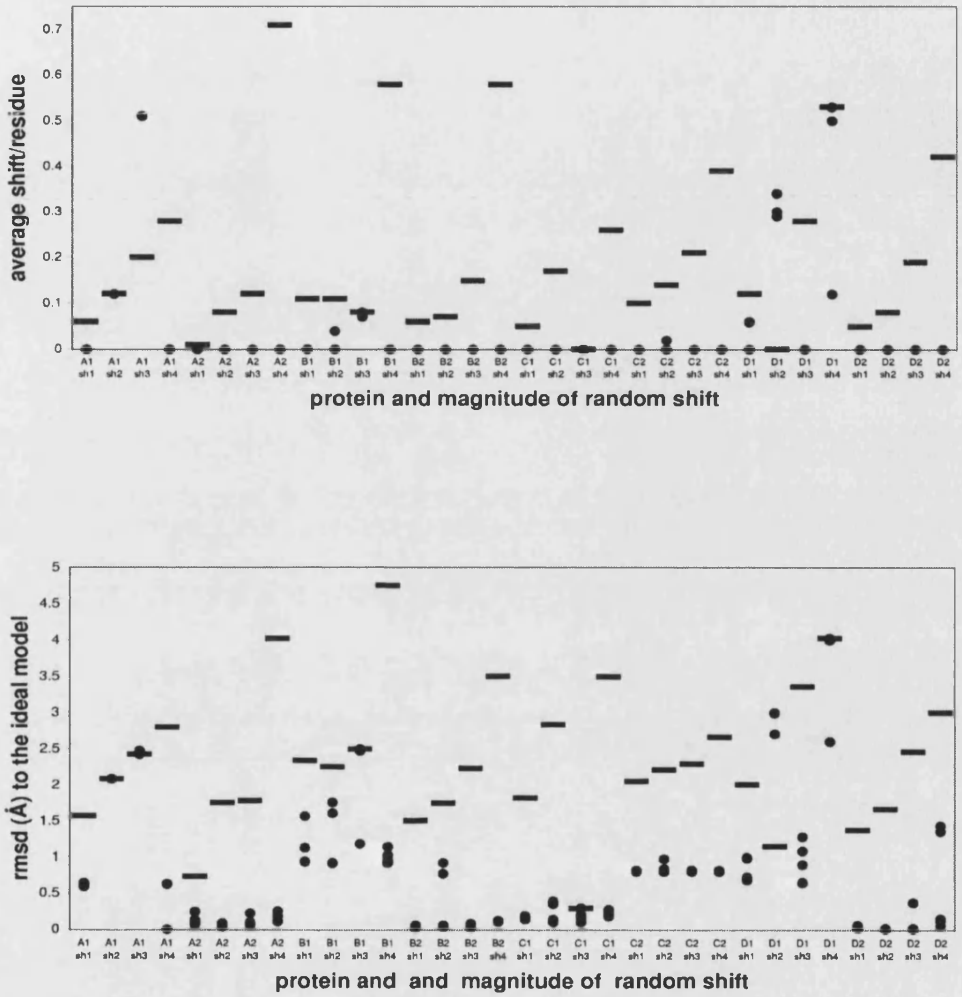


Figure 3.6: Protein recombination is able to generate optimal alignments and more accurate models from populations of models obtained from randomly shifted alignments. Eight populations of models (for sequences A1,A2,B1,B2,C1,C2,D1,D2) were created using randomly shifted alignments. Four different populations were generated for each sequence, using shifts of one, two, three and four positions. Each population was independently recombined five times. Final population averages (marked as periods) for each experiment are shown in the same column. Note that alignment shifts (on top) tend to disappear upon recombination with respect to the best initial model (marked as horizontal bars). RMSD differences to the known experimental structure (below) tend to diminish. Note also that some alignment errors cannot be recovered, such those found in populations A1sh2 or A1sh3,if there are no correctly aligned regions overlapping between parents.

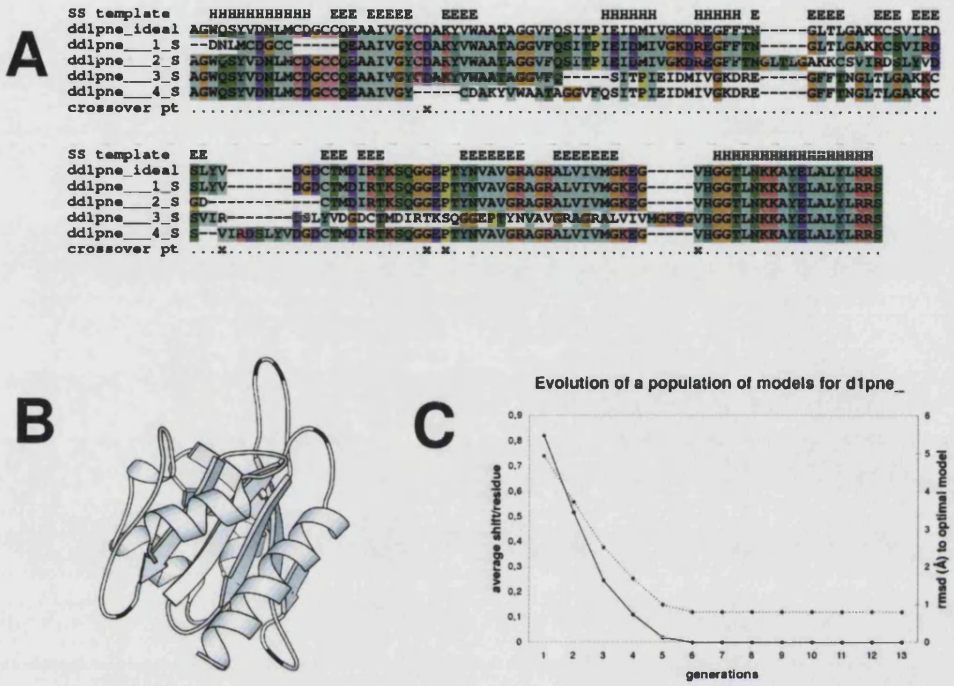


Figure 3.7: Protein recombination experiment in detail. **A)** Four shifted-alignment models (1_S,2_S,3_S and 4_S) for cow profilin (d1pne_ (Cedergren-Zeppezauer *et al.*, 1994)) were generated. Their $C\alpha$ RMSD to the correct conformation (ideal model) ranged from 2.3 to 9Å and their average alignment shift per residue from 0.16 to 2.68. The top row shows the STICK-assigned secondary structure for the ideal model (H for α-helix and E for β-sheet). The bottom row (x) marks the most frequent crossover points when these shifted models are recombined, recorded during the evolutionary process shown in C. **B)** shows the crossover points mapped (in black) on the final recombinant model, using the program Molscript (Kraulis, 1991). **C)** In this graph the average shift/residue (continuous line) and $C\alpha$ RMSD (dotted line) of the evolving population are plotted against the number of generations. After 13 generations, the simulation converged. The final population has an average $C\alpha$ RMSD of 0.8Å to the experimental d1pne_ structure (only 0.05Å to the backbone) and no alignment shifts.

Different templates and different alignments

The same previous eight SCOP domains (A1 to D2, see above) were used to set up a new recombination experiment. To build the initial populations of models we used single-template models built from alternative alignments (to the same template) simultaneously

with models built from a variety of templates in their SCOP families. The number of models used for these initial populations ranged from 10 to 102. In addition, to analyse how different several recombination runs can be, each initial population was used to start 10 independent experiments. Results are shown in Figure 3.8. The picture arising from this experiment is that alignment shifts are minimised upon recombination and can go beyond the best initial model in the population. At the same time, final population average RMSDs are comparable to the best initial model seeded. Furthermore these results pointed out the importance of replicating simulations for the same population to fully exploit the capability of the method. Since this is a population-based method, a population answer should be provided and this can be achieved by running independent simulations on the same input. Analysis of these experiments showed that on average RMSD differences between independent runs tend to be not significant, so they could be considered as ensembles of protein conformations, analogous to NMR structures.

Large-scale benchmark

To conclude the benchmark of the method, a large-scale protein recombination experiment was made on a set of 130 SCOP domains (27α, 38β, $26\alpha/\beta$ and $39\alpha+\beta$ folds). Domains were modelled using their family relatives as templates and only one sequence alignment per template. Due to computing time limitations, only one independent run was performed for each of the 130 populations. Despite this handicap, it turns out that the algorithm produces final populations of models that are comparable to the best initial model (see Table 3.5 and Figure 3.9) and that are consistently better than the initial population (around 1Å). In 92% of the cases (89% for models built from templates 40% or less identical in sequence) final population models are not significantly different to the best initial model. However, as expected from the reference experiment, using RMSD as a perfect fitness function, no improvement is seen beyond this limit. The good news is that the algorithm converges onto protein conformations close to the optimal model, suggesting that our method sorts templates better than sequence identity measures and that there is no need to select templates for modelling. The bad news is that more favourable protein conformations, according to the fitness function, do not always correspond to lower RMSD states (see Figure 3.10B for an example) and that, on average, the algorithm is not taking full advantage of the expected possibilities of combining different templates. To some extent this was predictable, since only one alignment per template was used for this experiment, making the method comparable to 3D-JIGSAW in that respect.

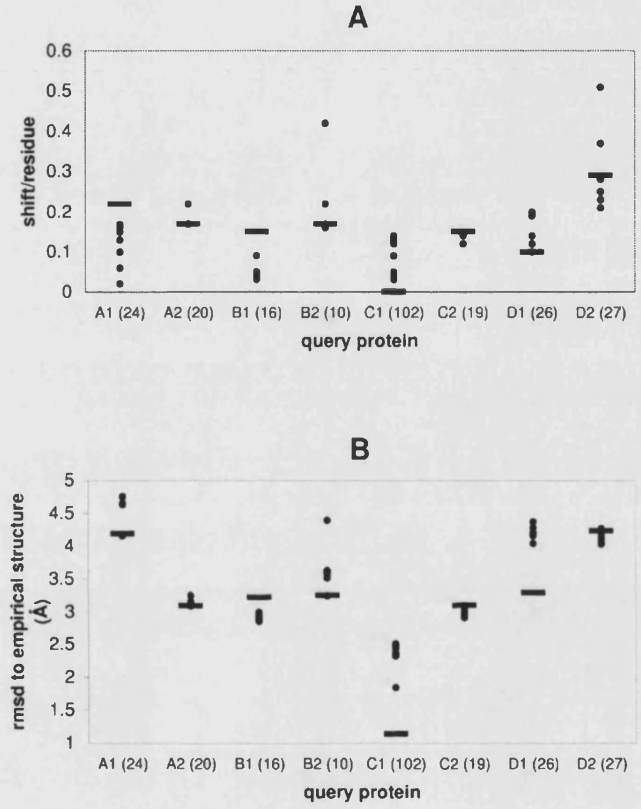


Figure 3.8: Alternative alignments and different templates may improve the performance of protein recombination. For each of eight model sets, ten recombination replications were produced, with final population averages shown in the same column. Note that alignment shifts (A) tend to diminish upon recombination with respect to the best initial model (marked with horizontal bars) if there is room for improvement. On the other hand, RMSD changes (B) are not equally consistent.

Analysis of RMSD changes

After recombining 130 sets of single-template models, only 3 final populations have conformations significantly better than the optimal template model (over 0.6Åof RMSD difference). Inspection of these models and others with minor improvements (30 recombi-

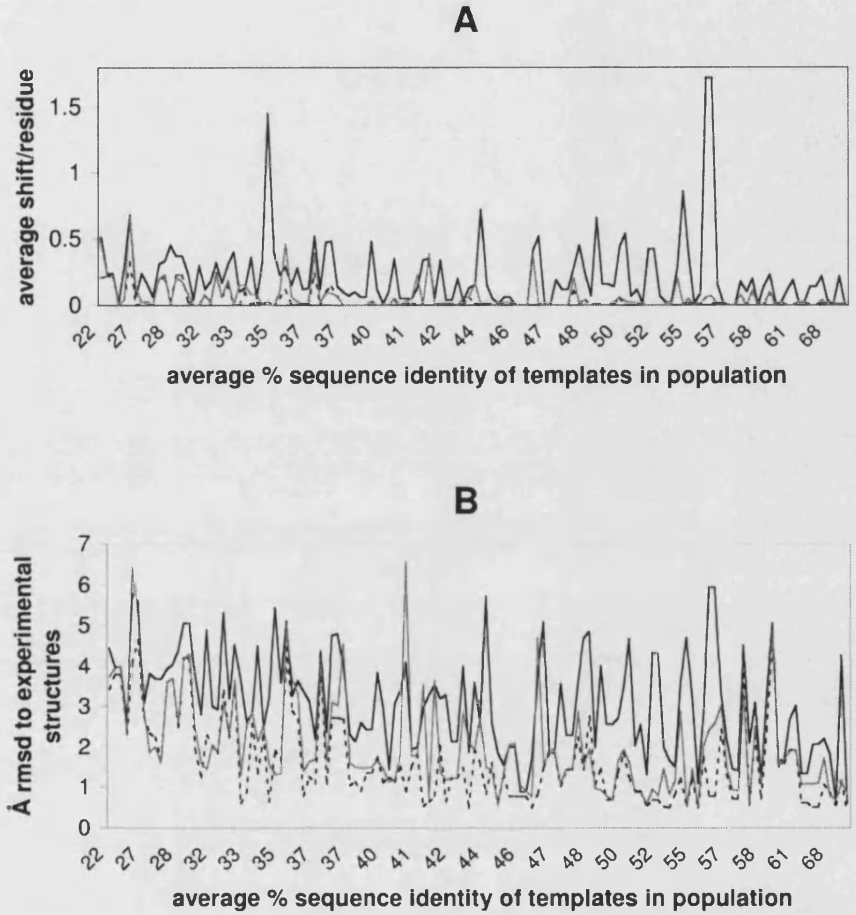


Figure 3.9: Performance of in silico protein recombination in a set of 130 unique experiments to model SCOP domains. Each model comes from a single sequence-aligned template. (A) Average population alignment shift measures are compared at the beginning (black solid line) and when the algorithm converges (grey solid line). Final populations of models are significantly better than initial (see Table 3.5) and the degree of improvement is limited by the best initial model (black dashed line) had we known it beforehand. (B) Average population RMSD to experimental structures for each SCOP domain is compared at the beginning and at the end of each experiment. Final population RMSDs are often over the best initial model, but differences are not significant in 120 out of 130 experiments (see Table 3.5).

nation experiments) shows that the improvements come from choosing alternative surface loop conformations or from small subdomain movements. Figure 3.10(A) shows one example in which the final population in the experiment achieved an RMSD value to the

case	$\Delta RMSD_{pop}$ (Å)	$\Delta RMSD_{best\ templ}$ (Å)	$\Delta shift_{pop}$ (shift/residue)	$\Delta shift_{best\ templ}$ (shift/residue)	generations
		Up to 100% identity (N=130)			
Best	-4.17	-0.88	-1.66	-0.18	11
Mean	-1.06	0.4*	-0.16	0.02	24
Worst	2.47	5.66	0.17	0.37	30
		Up to 40% identity (N=44)			
Best	-4.13	-0.88	-1.41	-0.18	12
Mean	-0.98	0.24*	-0.2	0.05	25
Worst	0.67	2.37	0.17	0.44	30+

Table 3.5: Benchmark of *in silico* protein recombination using our simple fitness function. Top: models using templates of any sequence identity. Below: only templates below 40% sequence identity had been used. The first column shows the final average population RMSD with respect to the founder populations RMSD values. The second column shows the evolution of RMSD with respect to the optimal template, had we identified it. Non-significant differences are marked with *. The third column shows the final average alignment shift with respect to the initial population. The fourth column highlights the same value now with respect to the best template. Finally, the last column shows the number of generations needed to reach convergence. Overall, in 92% of the simulation experiments the final population has an average RMSD to the experimental structure comparable to the model built from the best template, meaning that this method consistently identifies the best templates. If only the 40% subset is considered, the figure drops to 89%.

known structure of the protein that is significantly better (0.89Å) than the model built using the best template. In this case the improvement comes from the relative orientation of two subdomains from different templates that have been arranged together. Nevertheless, it is clear that on average models in populations do not improve their RMSD beyond the optimal template model. The value of this method is that it consistently converges around the optimal template's conformations, and these cannot be identified routinely.

Analysis of alignment accuracy

From these experiments it may be concluded that populations improve their average alignment shift (with respect to their structural alignment) through rounds of fitness selection and recombination. On average this improvement is about 0.16 shifts/residue (see Table 3.5), but the ceiling of this improvement is usually dictated by the optimal template model. Figure 3.11 shows how observed improvements in population energies correlate to average alignment shifts and RMSD changes through recombination experiments. A linear correlation between energy improvement and alignment shift change is found (Fig-

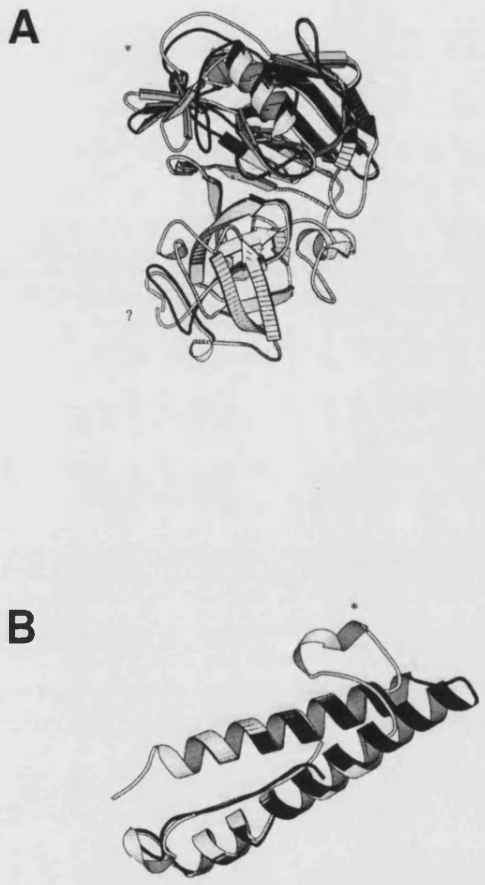


Figure 3.10: Limitations of the algorithm. Global RMSD improvements come usually from surface loop movements (usually intrinsically flexible) or small subdomain movements, as can be seen (A) in the experiment to model d1apr_ (mold acid protease 2APR (Suguna *et al.*, 1987)) from a population of 11 models built from different templates from the same SCOP family. The final population model is depicted in white, while the best initial model is shown in black (* points to the main differences observed comparing the two models and ? shows a broken loop, a common side-effect of protein recombination). The worst RMSD result obtained in our protein recombination benchmark is shown in B, where it was attempted to model d1dt0a1 (superoxide dismutase N-terminal domain in 1DT0 (Bond *et al.*, 2000)) from an initial population of 8 models. The simulation yields a final population RMSD of 5.35Å while the optimal template model (shown in black) is only 0.89Å away from the known experimental structure. In this particular example the long loop (*) is taken from a template (1MNG (Lah *et al.*, 1995)) whose crystallographic contacts bent the helical bundle.

ure 3.11A), but the interdependency between energy evolution and RMSD change (B) is less clear, and only tentatively can be approximated by a logarithmic function.

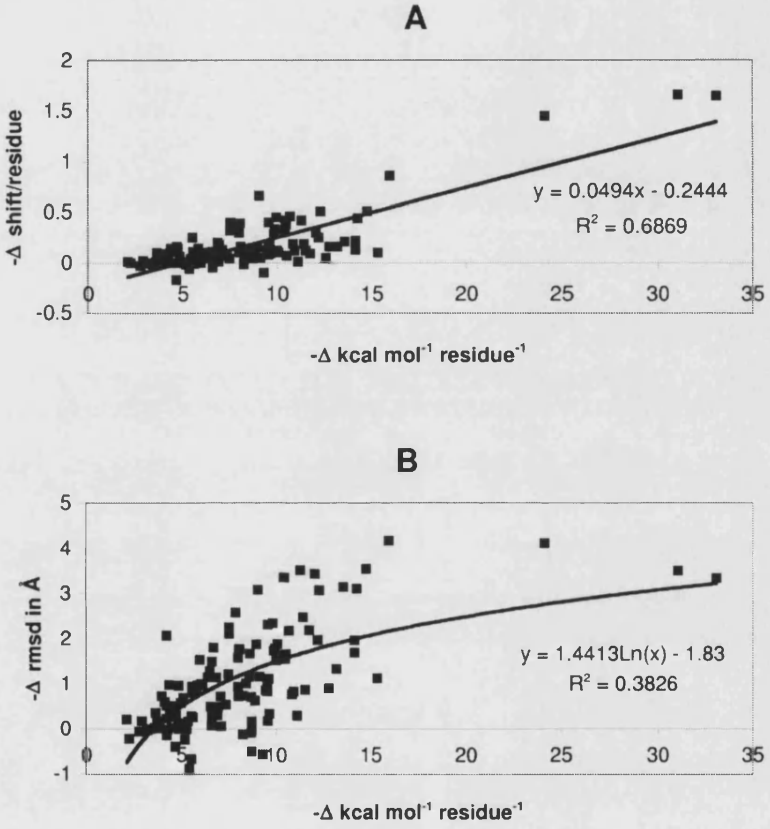


Figure 3.11: Correlations between energy improvements in populations and alignment and RMSD improvements calculated with data from 130 recombination experiments. (A) A linear correlation is found for the change in average alignment shift, suggesting that it could be predicted to some degree from the output of a recombination experiment. (B) The correlation to RMSD is weaker and only tentatively is modelled with a logarithmic function, suggesting that it would be of little value to predict RMSD improvements from energy profiles. A similar correlation is found when a linear function is tried.

3.6.3 Incorporating PSI-BLAST alignments

To compare our results to a standard alignment program, such as PSI-BLAST, the same previous set of 130 SCOP domains was re-investigated. Only templates found by PSI-BLAST, less than 40% identical to the query sequence, were used. Alignments were

taken directly from the program's output and subsequent models built using 3D-JIGSAW. These were added to models built using the same templates, but with our own alignments that consider secondary-structure information. The aim of the experiment was to compare the final population of recombined models to the PSI-BLAST-based model constructed from the alignment with the best *e*-value. The first observation from this experiment is that only 54 out of 130 domains can be modelled within these constraints, since templates for the remaining could not be found using default parameters. On this reduced dataset, recombined populations of models tend to be, on average, 0.51Å closer to the corresponding experimental molecular structure than the best *e*-value PSI-BLAST-based model. More importantly, the corresponding difference in alignment shift was on average 0.42 shifts/residue better than the PSI-BLAST model. However, in three cases, the recombination protocol did not improve beyond the PSI-BLAST alignment; indeed the original PSI-BLAST aligned models had better agreement with the experiment in some exposed loops. These results suggest that further improvements to the energy function can be made. This benchmark also suggested that simply taking the best *e*-value, and associated template, from a standard PSI-BLAST output, does not necessarily produce the best model. In addition, within these 54 examples, models built from the best *e*-value alignment were 0.81Å worse than the best models built from the ensemble of templates found by PSI-BLAST. In other words, *e*-values are not necessarily a good indication of how good a model will be, as suggested in Figure 3.1 for sequence identity. This observation also holds for alignment accuracy, since templates with lowest *e*-values produce models that are, on average, 0.58 shifts/residue displaced with respect to the best possible model.

3.6.4 Contribution of the solvation term

To further investigate the fitness function when applied to recombination experiments, the previously described 54 populations were reinvestigated to further assess the contribution of the solvation term. This was done by recombining these populations with and without this term in the energy function. The comparison of these simulations provides a clear conclusion: inclusion of the solvation term yields better recombinant models in terms of deviation to the experimental structures and alignments shift in 22 out of 54 domains; the remainder are very similar. On average, selecting models without the solvation term yields models that are 0.4Å worse than those selected including it. Alignments are further displaced by an average of 0.05 shifts/residue.

3.6.5 Discussion of results

The results presented here provide insights into two recurrent problems in protein comparative modelling: selecting templates and alignment errors. The novel methodology proposed here deals with both simultaneously, and despite some deficiencies it is found to be robust to some alignment errors. It also confidently classifies possible protein conformations for a given sequence based on its homologous partners in the structural database, the templates. These two features are crucial to automate the construction of comparative models. Nevertheless, comparison of the fitness function with the ideal (Tables 3.4 and 3.5) suggests that further improvements can be made to this function. Some limitations and applications of this algorithm are now discussed.

Applications

As shown in the analysis of the results, the method presented here can improve the alignment accuracy of comparative models and avoids the step of selecting templates, since models from all possible templates can be used. If these models are to be used as guides for site-directed mutagenesis experiments, one of the most popular applications for CM (Marti-Renom *et al.*, 2000), alignment accuracy is essential to target the correct residues. Comparative models have also been applied to fit protein structures into electron microscopy density maps of single molecules or supramolecular complexes (Zhang *et al.*, 2000; Wriggers & Chacon, 2001; Elcock, 2002), and alignment accuracy is therefore important to place the corresponding protein parts into the experimental data. A different application of modelling, at the population level, would be to gain insights into fold flexibility within a given molecule or even across families, because members of the same population of models can have geometrical differences that cannot be penalised at the level of fitness. This could simply be pointing out the weakness of the fitness function used, but recent work (Koehl & Levitt, 1999; Zagrovic *et al.*, 2002a) using different functions and different approaches, such as the Metropolis rule, propose that sequence or structure ensembles represent more faithfully the nature of a given protein fold. The most important feature of this methodology is its ability to recover alignment errors and to generate different alignments from those contained in the initial population. This could be used to combine comparative models obtained from different sources, templates and alignments to get, not a consensus answer (something other programs already do (Lundstrom *et al.*, 2001; Ginalski *et al.*, 2003)) but a model close to the optimal template that could correct alignment errors found in the initial population.

Limitations

The presented algorithm has several limitations, the most obvious being the fitness function. Improvements to it will be translated into improvements of the algorithm performance, within the limits defined in our benchmark using RMSD as an ideal fitness function. This means that the algorithm can potentially take advantage of better fitness functions found by the community in the future or those already described in the literature,(see for example Holm & Sander (1992); Koehl & Levitt (1999); Janardhan & Vajda (1998); Keasar & Levitt (2003)). In particular, it seems important to explicitly include terms accounting for the formation of hydrogen bonds. However, better functions may require more computing time, limiting their practical applicability. In addition, because the algorithm creates new protein conformations every generation by "cut and paste", if finer energy functions were used, it would be necessary to minimise protein conformation energies every generation, adding yet more computational overhead to the process. The fitness function used for this work is fast to calculate but at the price of being less accurate. This has the benefit that population members need not be minimised every generation. Despite this, protein recombination experiments can still take several hours in a worst-case scenario (see Section 3.11). In a practical situation, models generated by *in silico* protein recombination often need to be minimised, particularly to fix broken loops. In general, the energy function used and the run-time checks (see Section 3.5.1) are sufficient to produce models with minor stereochemical problems that can be fixed with a subsequent full-atom minimisation algorithm. The second limitation of the method is the search for meaningful alternative alignments to the modelling templates. We have shown the ability of the method to recover from some alignment errors and to improve the population alignment accuracy, with the condition that partially correct alignments are present in the initial population. If all the initial alignments in a particular region are wrong, the method would not be able to provide an accurate conformation for that part of the protein. This suggests that models used for recombination experiments should cover many different but reasonable alignment possibilities. Unfortunately the total number of possible sequence alignments is vast and no hint can be given about the minimal alignment set required to solve the problem. Suboptimal alignment strategies, like the one used in our experiments (Saqi & Sternberg, 1991; Saqi *et al.*, 1992), and different alignment procedures could be used, since it is accepted that different sequence alignment tools usually give different answers to the same non-trivial alignment problem and often each of them would give optimal alignments in particular cases but not in others, as mentioned in 2.2.1. Finally, the stochastic nature of the algorithm implies that slightly different answers for the same

input can be obtained. This can be utilised to provide useful information concerning fold flexibility, as discussed above, but would of course require additional computing time.

The role of mutation

One of the conclusions of this benchmark is the secondary role of our mutation operator, compared to recombination, in generating useful conformation variability. This in theory undermines the capacity of the method to generate novel protein conformations, substantially different to any of the templates used. Of course this is related to the way the mutation mechanism is implemented, and because the current method is simply an averaging procedure, with no attempt to correct generated distorted side-chains, we believe it is possible to increase the contribution of mutation. It would imply quality checks after averaging or, as with SWISS-MODEL (Guex *et al.*, 1999), averaging only the Cα atoms and then reconstructing the rest of the residue. To test if variability generated by other means could improve the performance of the method, a small recombination experiment was carried out in which the original sets of initial models were used to generate extra compatible protein conformations using the method CONCOORD (de Groot *et al.*, 1997). The results were not significantly different, so we concluded that mutation, in this context and with this fitness function, is secondary to recombination. Similar observations have been made in related contexts (Xia & Levitt, 2002).

Crossover and secondary structure elements

An important feature of the method is the choice of crossover points between models. In this initial implementation, crossover is only permitted to occur out of regular secondary structure elements, as defined by STICK, a program that assigns secondary structure states based on vectors that represent the topology of the fold. The reason for this is that loops seemed to be the natural place to cut and paste peptides. Furthermore, we prefer not to recombine in helices or strands to conserve their native geometry and to avoid additional efforts to reconstruct them. However, there is no reason to believe that genetic recombination, to which this algorithm is analogous to, occurs only outside of DNA regions coding for regular secondary structure elements.

3.7 CASP5 benchmark

Around the time we were benchmarking these modelling procedures and analysing the results, the prediction season of CASP5 started, towards the end of May 2002. A total of

187 research groups from around the world and up to 72 web servers registered to submit structural predictions for 67 proteins, shown in Table 3.6. The broad goals of CASP experiments, summarised in Section 1.5, are to address the following points:

1 Are the models produced similar to the corresponding experimental structure?

2 Have similar structures that a model can be based on been identified?

3 Is the mapping of the target sequence onto the proposed structure (i.e. the alignment) correct?

4 Are the details of the models correct?

5 Has there been progress from the earlier CASPs?

6 What methods are most effective?

7 Where can future effort be most productively focused?

As in previous CASPs, independent assessors would evaluate the predictions, emphasising primarily on the effectiveness of different methods. The experiment concludes with the CASP5 meeting to discuss progress and relative performances of each method. The part of CASP that deals exclusively with automatic methods is called CAFASP (Critical Assessment for Fully Automated Structure Prediction).

Table 3.6: List of targets included in CASP5, including the published experimental structures added to the PDB as of May 2003. The reader can browse through the targets and the models submitted by the participating groups at the CASP5 website (http://predictioncenter.llnl.gov/casp5).

Target-id	Name	residues	Exp.Method	Description
T0129	HI0817	182	X-ray	H. influenzae
T0130	HI0073	114	X-ray	H. influenzae
T0131	HI0857	100	X-ray	H. influenzae
T0132	HI0827	154	X-ray	H. influenzae
T0133	HIP1R	312	X-ray	N-terminal domain, rat
T0134	AP3DELTA	251	X-ray	Delta-adaptin appendage domain, human
T0135	BSPA	108	X-ray	Boiling stable protein, P.tremula

(continued on next page)

Table 3.6:

(continued from previous page)

Target-id	Name	residues	Exp.Method	Description
T0136	TC12S	523	X-ray	Transcarboxylase 12S subunit, P.shermanii (PDB 1ON3 and 1ON9, (Hall *et al.*, 2003))
T0137	FABP1	133	X-ray	Fatty acid binding protein FABP1, E.granulosus
T0138	KaiA135N	135	NMR	N-terminal domain, S. elongatus (PDB 1M2E and 1M2F, (Williams *et al.*, 2002))
T0139	DFF-C	83	NMR	Caspase Associated DNase domain (225-307), human (PDB 1KOY (Fukushima *et al.*, 2002))
T0140	1B11	103	X-ray	synthetic protein
T0141	AmpD	187	NMR	C. freundii (PDB 1IYA (Liepinsh *et al.*, 2003))
T0142	NITRO	282	X-ray	Nitrophorin, C.lectularius
T0143	V8prot	216	X-ray	V8 protease, S.aureus
T0144	CYP	172	X-ray	Cyp protein, L.luteus
T0145	GLI	216	X-ray	Gliotactin C-terminus portion, D.melanogaster
T0146	ygfZ	325	X-ray	E.coli
T0147	ycdX	245	X-ray	E.coli (PDB 1M65 and 1M68 (Teplyakov *et al.*, 2003))
T0148	HI1034	163	X-ray	H.influenzae
T0149	yjiA	318	X-ray	E.coli
T0150	L30E	102	X-ray	Ribososmal protein L30E, T. celer (PDB 1H7M (Chen *et al.*, 2003))
T0151	SSBP	164	X-ray	Single-strand binding protein (SSB), M.tuberculosis H37Rv

(continued on next page)

Table 3.6:

(continued from previous page)

Target-id	Name	residues	Exp.Method	Description
T0152	Rv1347c	210	X-ray	Hypothetical protein Rv1347c, M.tuberculosis H37Rv
T0153	DUT	154	X-ray	Deoxyuridine 5'-triphosphatenucleotidohydrolase (dUTPase),M.tuberculosis
T0154	PANC	309	X-ray	Pantothenate synthetase, M.tuberculosis (PDB 1MOP (Wang & Eisenberg, 2003))
T0155	FOLX	133	X-ray	Probable dihydro-neopterin aldolase (DHNA),M.tuberculosis
T0156	MENG	157	X-ray	S-adenosylmethionine:2-demethylmenaquinone methyltransferase, M. tuber-culosis
T0157	yqgF	138	X-ray	E.coli
T0158	AES	319	X-ray	Acetyl esterase, E.coli
T0159	PROX	309	X-ray	Glycine betaine-binding periplasmic protein, E.coli
T0160	VAP-A	128	X-ray	VAP-A protein, rat
T0161	HI1480	156	X-ray	HI1480, H.influenzae
T0162	CAZ1	286	X-ray	F-actin capping protein alpha-1 subunit, chicken (PDB 1IZN (Yamashita *et al.*, 2003))
T0163	GLOX	369	X-ray	Glycin oxidase, B.subtilis
T0164	C20	166	X-ray	C20, chicken
T0165	CAH	318	X-ray	Cephalosporin C deacetylase, B. subtilis

(continued on next page)

Table 3.6:

(continued from previous page)

Target-id	Name	residues	Exp.Method	Description
T0166	SLYA	150	X-ray	Transcriptional regulator SLYA, E. faecalis
T0167	yckF	185	X-ray	Hypothetical Cytosolic Protein B.subtilis
T0168	GLS2	327	X-ray	Glutaminase, B.subtili
T0169	yqjY	156	X-ray	B.subtilis (PDB 1MK4 (Zhang *et al.*, 2002))
T0170	HYPA	69	NMR	FF domain of HYPA/FBP11, human (PDB 1H40 (Allen *et al.*, 2002))
T0171	BIOH	256	X-ray	Protein BioH, E.coli (PDB 1M33 (Sanishvili *et al.*, 2003))
T0172	MRAW	299	X-ray	Conserved hypothetical protein, T.maritima (PDB 1M6Y and 1N2X (Miller *et al.*, 2003))
T0173	Rv1170	303	X-ray	Mycothiol deacetylase, M.tuberculosis
T0174	XOl-1	417	X-ray	Protein XOl-1, C. elegans (PDB 1MG7 (Luz *et al.*, 2003))
T0175	yjhP	248	X-ray	Hypothetical protein yjhP, E.coli
T0176	yggU	100	NMR	Hypothetical protein yggU, E.coli (PDB 1N91 (Aramini *et al.*, 2003))
T0177	HP0162	240	X-ray	Hypothetical protein HP0162, H.pylori
T0178	DEOC	219	X-ray	Deoxyribose-phosphate aldolase, A.aeolicus

(continued on next page)

Table 3.6:

(continued from previous page)

Target-id	Name	residues	Exp.Method	Description
T0179	ywhF	276	X-ray	Spermidine synthase homolog, B.subtilis
T0180	MTH467	53	NMR	Hypothetical protein MTH467, M.thermoautotrophicum
T0181	YHO7	111	NMR	Hypothetical protein YHR087w, S.cerevisiae
T0182	TM1478	250	X-ray	T.maritima
T0183	TM1559	248	X-ray	T.maritima
T0184	TM1102	240	X-ray	T.maritima
T0185	TM0231	457	X-ray	T.maritima
T0186	TM0814	364	X-ray	T. maritima
T0187	TM1585	417	X-ray	T. maritima
T0188	TM1816	124	X-ray	T. maritima
T0189	TM0828	319	X-ray	T. maritima
T0190	YEDX	114	X-ray	Transthyretin-related protein, E.coli
T0191	AROE	282	X-ray	Shikimate 5-dehydrogenase, M.jannaschii (PDB 1NVT (Padyana & Burley, 2003))
T0192	SSAT	171	X-ray	Spermidime/Spermine Acetyltransferase (SSAT), human
T0193	ATBP	211	X-ray	AT-rich DNA binding protein (ATBP), T.aquaticus
T0194	Y450	237	X-ray	Conserved hypothetical protein, M.pneumoniae
T0195	YJG8	299	X-ray	Hypothetical esterase in SMC3-MRPL8 intergenic region, S. cerevisiae

To evaluate the results, organisers and assessors split the targets into structural domains and classify each of them according to their prediction difficulty, using sequence and structural similarity criteria. CM targets are considered the easiest, whilst NF are the most difficult:

category	full name	methods needed to find optimal template(s)
CM	Comparative Modelling	BLAST and up to five iterations of PSI-BLAST with e-value$<$ 0.005.
CM/FR	CM/Fold Recognition	Transitive PSI-BLAST with e-value$<$ 0.02, in which hits found in initial searches are fed back into PSI-BLAST to find remote templates.
FR(H)	FR (Homology)	Sequence similarity found when query and templates are structurally aligned.
FR(A)	FR (Analogy)	No sequence similarity found when query and templates are structurally aligned.
NF	New Fold	no templates found in the PDB.

Table 3.7: Classification of CASP5 targets by expected prediction difficulty.

The two main numerical criteria used by the assessors to compare and evaluate predictions of tertiary structure are the following scores:

- GDT_TS, the average of the maximum number of residues that can be superimposed between the experimental structure of the target and a predicted model within 1Å,2Å,4Å and 8Å distances in a sequence-dependent manner.
- AL_4, number of residues in a model for which the corresponding residue in the experimental structure is within +/-4 residues of the correct one (shifted up to four residues) and the distance in between is less than 4.0Å. This is a sequence independent measure.

Both scores are percentages and can be calculated using the program LGA by Zemla (2003), member of the CASP organising team. These scores fulfil the need for a unique and objective numerical analysis of the results by all predictors.

3.7.1 Our protocol for CASP5

As in previous CASP experiments, there were basically two categories of targets: those for which one or more templates can be found and those for which no obvious templates can

be assigned. We tried to model each of the 67 targets on the assumption that there was at least one template within the PDB. Therefore the only differences between the CM and FR targets would be the more or less sophisticated tools used to find and align templates. The rest of the modelling procedure for these two categories would be essentially the same. Indeed, the assessors for these categories in CASP4 pointed out that template selection and sequence alignment errors remained as the main problems affecting the quality of models (Tramontano *et al.*, 2001; Sippl *et al.*, 2001). For these reasons, we decided to use the same tools and strategies for all CASP5 targets. In our hands, FR and CM are the same problem, only the sequence similarities involved are of different magnitude.

The underlying assumption we had was that a combination of alignment methods should be better than any individual method and that there is currently no way to confidently identify the best template and therefore several templates should be used and combined. Our approach, as has been presented, tackles both problems simultaneously. The idea is that different templates for a given target are just different possible structures for the same sequence. All templates are assumed to be homologous proteins, synthesised from homologous genes, that can undergo genetic recombination or mutation. Since a model can be considered as an alignment in three dimensions, models for alternative alignments to the same template can be used. This simple principle was implemented and applied to all CASP5 targets. The protocol can also be followed in Figure 3.12.

Initial population of models. Initially, the web server DomainFishing was used to define protein domains within each target sequence and to find suitable modelling templates. Resulting alignments were inspected and corrected if suspected to be incorrect (a variety of biochemical and subjective knowledge-based criteria were used here). When found, different alignments to the same template were added to the pool. In several cases, such as T0130, annotations from the templates or their corresponding PFAM families were used to check the correctness of the alignment in active/binding sites. In cases where DomainFishing returned no templates, alignments were generated using a pssm-pssm search against a non-redundant PDB library (coded by Paul A.Bates). As in DomainFishing, this program calculates up to seven different alignments for each library member. Models from these alignments were built using 3D-JIGSAW, using the interactive mode to edit the alignments.

To gain extra variability in sequence alignments, templates and alternate loop conformations, models were also taken from different CAFASP servers that return full atomic coordinates. These were: FAMS(Ogata & Umeyama, 2000), EsyPred3D(Lambert *et al.*, 2002), Arby(Fraunhofer-Institute), Alax , Robetta (Simons *et al.*, 1997b) and Pmodeller

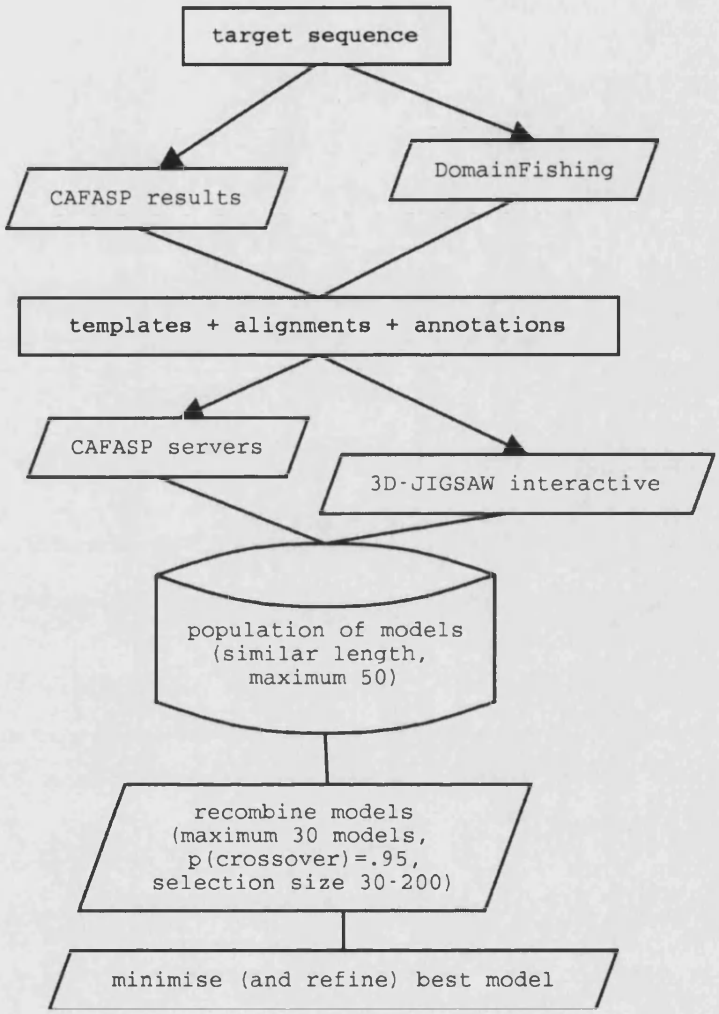


Figure 3.12: Our modelling protocol during CASP5.

(Lundstrom *et al.*, 2001; Wallner & Elofsson, 2003). In cases where the fold of the target was not clear, models built using the most popular templates from the most popular SCOP superfamilies were preferred.

Models were inspected and missing parts, typically loops, added using in-house software written by Paul.W.Fitzjohn before going to the next step. In essence, this software explores ϕ/ψ space to allow a peptide (the missing loop) to connect a gap in a protein fold. Models were then energy-minimised in order to smooth their ϕ/ψ geometry and to permit unbiased energy calculations at later stages.

Growing the population by recombination and mutation. The protein recombination procedure explained previously was applied here with recombination rates of 0.95, since we knew that out implementation of mutation was not helping much.

Selecting the best proportion. When a population reaches the upper limit (between 2 and 4 times its initial size, 30 to 200 models in our CASP5 simulations), members are ranked according to their fitness. To assure that quality models are not lost prematurely, the selection rate was kept at 25%, the value we had been testing in our internal benchmarks.

Convergence and final refinements. When all members of the population have converged to a similar energy, there is no room for further generation of variability and the evolution process stops. In most cases this final population consists of several representatives of the same protein conformation with average backbone deviations in the order of 0.1Å, but sometimes substantially different subpopulations can be obtained. One of these representatives, usually the first or the most populated, was taken as the final model and carefully inspected, using the program Quanta (Accelrys Inc.), to detect unfavourable peptide conformations, check the Ramachandran plot and a final energy minimisation using the CHARMM22 force field, after adding covalent hydrogens and protons to acid and basic groups, assuming neutral pH. No cofactors were considered. This procedure is able to fix distorted side-chains generated by mutation, particularly twisted cyclic groups or elongated bonds. At this point we had a CASP5 unrefined model. For targets T0134, T0165, T0177 and T0185 a further refinement step was performed, consisting of running an all-atom molecular dynamics simulation. This is explained in detail in Section 3.8.

3.7.2 CASP5 results and analysis for FR/NF targets

All 67 CASP5 targets were modelled using the above protocol. This population approach was used as an attempt to optimise template-based models obtained from different sources. The analysis of the results shows that, in general, recombined models are not significantly different to the best initial model, if that could have been identified at the time of submission. This is the same message we learnt from out internal benchmark. Only in a handful of cases did recombination yield slightly better models. With a similar frequency, the algorithm yields slightly worse models than the best initial, particularly when all the initial models are poor. The performance of the method is similar across all CASP5 targets, but here only remote homology targets, down to the New Fold (NF) cate-

gory, are discussed, as alignment errors and incorrect selection of templates become more frequent for these targets. Indeed the assessor for the FR category (Nick Grishin) invited us to present our results for these (Contreras-Moreira *et al.*, 2003b). Table 3.8 shows our analysis for the results of these 24 protein domains, after comparing our models with the targets for which the experimental structure is available. As described in the previous Section, a set of template-based models was constructed for each target to seed the initial population for a recombination experiment. The final model submitted was selected from those in the last generation of models, after convergence. This table shows how different the final recombinant models (Rec) are with respect to the initial models, constructed using the servers stated on the top of each column. To compare models, the standard CASP scores were used, AL_4 and GDT_TS.

Table 3.8: Performance of protein recombination in the CM/FR, FR(Homology), FR(Analogy) and FR/NF categories. The first column states the target name (*w* is shown on targets modelled using templates with incorrect folds). The left side of the table shows AL_4 scores for the initial models fed into the recombination algorithm. These models were obtained from different web-servers (3D-JIGSAW, Pmodeller and Others). Ranges show the best and the worst scored models, with the total number of models in square brackets. The fifth column shows the AL_4 score for the recombinant models. The right side of the table shows the analysis of the same data, using GDT_TS scores. See the main text for the definitions of these scores. "Others" are servers participating in CAFASP3, where * indicates servers Fams,Alax,Robetta, ? Robetta, + Robetta,Arby, # Fams and $ Fams,Alax. Finally, ! indicates a FR method by secondary structure pattern matching, developed by P.W.Fitzjohn in our lab.

	AL_4				GDT_TS			
	3D-Jigsaw	Pmodeller	Others	Rec	3D-Jigsaw	Pmodeller	Others	Rec
				CM/FR				
T0130	61-60[2]			63	43.2-40[2]			37.3
T0132	66.4-56.8[3]	84.9-56.8[2]		82.2	42.3-39.4[3]	60.4-44.3[2]		61.6
T0159_1		53.3-18.6[10]	26.9-13.2[3]	40.7		36.9-16.2[10]	17-12.6[3]*	25.4
T0159_2		53.5-37.3[10]	44.4-32.4[3]	52.8		34.3-23.4[10]	27.8-23.4[3]*	33.1
T0168_1	58.8-49.4[4]	65.9-43.5[10]		53.5	40.1-34.8[4]	42.8-30.4[10]		35.7
T0168_2	26.2-17.7[4]	31.2-16.3[10]		16.3	22.1-19.1[4]	24.2-18.4[10]		19.7
				FR(H)				
T0134_1	67.5[1]	72.2-32.5[7]		69.8	40.7[1]	43.8-20.4[7]		39.1
T0134_2	89.6[1]	87.7-70.7[7]		82.1	58.5[1]	66-42.7[7]		63.4
T0138	78.5-15.6[6]	83.7-60.7[10]		66	43.5-12.4[6]	58.3-47.9[10]		48.7
T0157		80.8-30.8[8]	41.7-10.8[4]	74.2		56.4-25[8]	56.4-22.9[4]?	52.5
T0174_1		15.2[2]		16.7		14.2[2]		14.5
T0174_2		23.9[2]		26.4		23.7[2]		23.7
				FR(A)				
T0135(*w*)			25.5[1]				17.4[1]!	
T0147		22.6-14.5[5]	27.8-20.5[2]	43.6		32.9-23.9[5]	29.6-27.1[2]+	27.7
T0148_1	5.6[1]	23.9-5.6[5]		64.8	27.5[1]	45.1-26.8[5]		45.8
T0148_2	13.2[1]	13.2-6.6[5]		27.5	24.7[1]	35.7-28[5]		29.7
T0187_2(*w*)		15-8.8[2]		17.1		11.8-10.6[2]#		11.9
T0191_1(*w*)	15.1[1]	49.6-12.2[8]	21.6-12.2[5]	15.8	14.9[1]	34.9-15.3[8]	18-14.9[5]#	16.4
T0191_2	80.4[1]	81.8-61.5[8]	83.9-60.1[5]	80.4	51.6[1]	56.3-40[8]	52.8-43.4[5]#	52.6
				FR/NF				
T0170		63.8-13[10]		47.8		49.6-31.9[10]		37.7
T0172_2		36.6-17.8[4]	26.7-14.8[11]	17.8		24.7-19.8[4]	20.5-17[11]$	18.1
T0173		18.1-14.6[3]	19.9[1]	18.1		13-10.1[3]	15.1[1]#	13
T0186_3	36.1-30.6[3]	50-30.6[10]	44.4-33.3[5]	38.9	29.2-27.8[3]	36.8-30.6[10]	29.9-28.5[5]#	29.9
T0187_1			17.6-16[2]	18.2			17.5-16.6[2]#	18.2

Some particular examples for each category, as assigned by the assessors, are now analysed in more detail.

3.7.3 T0132 (HI0827, *Haemophilus influenzae*)

This CM/FR target was identified as a thioesterase by DomainFishing. Using profile-profile searches (see Section 3.7.1) the template 1BVQ, a CoA-thioesterase from *Pseudomonas sp.*, was confidently found (with 16% of sequence identity). However, the alignment was not trivial, so three different alignments were used to build models with 3D-JIGSAW and two more models were taken from Pmodeller, one of them using a different template, 1C8U, another bacterial CoA-related enzyme. Recombination built a model that incorporated fragments from both templates but eventually had a very similar score to the best initial model, a Pmodeller model based on an alignment generated by INBGU(Fischer, 2000). We now analyse the major difficulties of the model, the phasing of strands 2 and 5 of the core β-sheet. For strand 2, our initial set of five alignments contained only segments shifted 1 or 2 positions with respect to the correct alignment. The resulting recombinant alignment is shifted 1 position at this point. However, for strand 5 there were two initial correct alignments (the remaining alignments were shifted by 1 and 2 positions) and they were incorporated into the final recombinant model. These results show how important it is to properly sample segments of ambiguous alignment, as the algorithm cannot generate alignments omitted from the initial population. The *msuper* structural alignment of the five initial models, the recombinant models and the experimental structure of T0132 is shown in Table 3.9.

3.7.4 T0157 (yqgF, *Escherichia coli*)

This target was classified as FR(Homology) by the CASP5 assessors and was related to DNA binding proteins according to the homologous sequences found by PSI-BLAST in the NCBI nr database. We could not find any confident template(s), so we took models from the CAFASP3 results page. In particular, models from Robetta and Pmodeller were selected as they used the most popular templates (1KCF and 1HJR, *E.coli* and yeast endonucleases, 17% sequence identity). Different alignments were found for each of them and a recombination experiment was set to select the best. The recombinant model is comparable, though slightly worse than, the best initial one (based on an alignment generated by FUGUE (Shi *et al.*, 2001) using 1HJR), but incorporating two different loops and a differently phased α-helix. The main difficulty of the target, an α-helix with a different angle to equivalent helices on the templates, was not resolved. As of June 2003, this structure has not as yet been released.

```
                .....str1.....
SS              CCCCCCCCCCEEEEEEECCHHHCCCCCCCCHHHHHHHHHHHHHHHH-------H----HH
real_T0132      TDKNGRQSKGVLLLRTLAMPSDTNANGDIFGGWIMSQMDMGGAILA-------K----EI
Rec             -------KGVLL-LRTLAMPSDTNANGDIFGGWIMSQMDMGGAILA------K---EI-A
1bvq0           -------KGVLL-LRTLAMPSDTNANGDIFGGWIMSQMDMGG---AI-----LA--KEIA
1bvq1           -------KGVLL-LRTLAMPSDTNANGDIFGGWIMSQMDMGGAILA------K---EI-A
1bvq2           -------KGVLL-LRTLAMPSDTNANGDIFGGWIMSQMDMGGAI-L------A---KEIA
Pmod0           -------KGVLL-LRTLAMPSDTNANGDIFGGWIMSQMDMGGAILA------K---EI-A
Pmod4           -------ANFTD-KNGRQSKGVLLLRTLAMPSDTNANGDIFGGWIMSQMDMGGAILAKEI
Rec_shift       -------11111-00000000000000000000000000000000------1---22-3

                ......str2..................str3.........str4.............
SS              HCCC-EEEEEECCCCCCC-C-CCCCEEEEEEEEEEECCCEEEEEEEEEE-CC-CCCC-
real_T0132      AHGR-VVTVAVESMNFIKP-I-SVGDVVCCYGQCLKVGRSSIKIKVEVWVK-KV-ASEP-
Rec             H-GRVVTVAVES-MNFIKP-ISV-GDVVCCYGQCLKVGRSSIKIKVEVWVK----KVASE
1bvq0           HGRVVTVAV-ES-MNFIKPISVG-DVVCCYGQCLKVGRSSIKIKVEVWVKK----VA---
1bvq1           HGRVVTVAV-ES-MNFIKPISVG-DVVCCYGQCLKVGRSSIKIKVEVWVKK----VA---
1bvq2           HGRVVTVAV-ES-MNFIKPISVG-DVVCCYGQCLKVGRSSIKIKVEVWVKK--V---A--
Pmod0           H-GRVVTVAVES-MNFIKP-ISV-GDVVCCYGQCLKVGRSSIKIKVEVWVKKVASEPI--
Pmod4           AHGRVVTVAVES-MNFIKP-ISV-GDVVCCYGQCLKVGRSSIKIKVEVWVK----KVASE
Rec_shift       1-0011111111-000000-011-00000000000000000000000000----33222

                .......str5.....................................
SS              C-CCEEEEEEEEE-EEECCCC---CCCCC-CCCCCCHHHHHHHHHHHC
real_T0132      I-GERYCVTDAVFT-FVAVDNN---GRSRT-IPRENNQELEKALALISE
Rec             PIGERYCVTDAVFT-FVAVDNNGRSRTIPR-ENN-QE--LEKALALI--
1bvq0           S-EPIGERYCVTDAVFTFVAVD-NNGRSR--------------------
1bvq1           SEPIGERYCVTDAV-FTFVAVDNNGRSRT--------------------
1bvq2           SEPIGERYCVTDAV-FTFVAVD-NNGRSR--------------------
Pmod0           --GERYCVTDAVFT-FVAVDN-NGRSRTIPRENN-QE--LEKA------
Pmod4           PIGERYCVTDAVFT-FVAVDNNGRSRTIPR-ENN-QE--LEKA------
Rec_shift       21000000000000-000000033333333-333-22--00000000--
```

Table 3.9: Structural alignment of models for T0132. Three-state DSSP-assigned secondary structure is shown on top, then the experimental structure, the recombinant model, three 3D-JIGSAW 1BVQ models and two Pmodeller models. β-strands are numbered on the top row. On the bottom row of each block the alignment shift per residue of the recombinant model is shown, where 0 means a correctly aligned residue, 1 a one residue shift and so on. Note that all the alignments for strand 2 are shifted 1 or 2 positions; for strand 5 the right alignment was correctly chosen among the three different possibilities. The 3D structure cannot be shown since it has not been published yet, as of June 2003.

3.7.5 T0147 (ycdX, *Escherichia coli*)

This FR(Analogy) target was identified as a PHP (polymerase and histidinol phosphatase) domain by DomainFishing. This superfamily includes several types of DNA polymerases, histidinol phosphatases, and a number of uncharacterised protein families. These proteins have four conserved sequence motifs that contain invariant Histidine and Aspartate residues implicated in metal ion coordination (Teplyakov *et al.*, 2003). No confident template could be found using our own set of tools, so once again models for the most popular templates found by the CAFASP servers were downloaded. All these templates (1DHP,1H5Y,1QO2,1THF,1NAL) were TIM barrels, with 8 β-strands, whilst the target sequence had only 7 β-strands predicted. No conclusive functional hint was found to help in selecting templates, so a set of 7 models from Pmodeller, Robetta and Arby was

recombined using the genetic algorithm. The final recombinant model selected by our fitness function has a poor GDT_TS score, as also do the initial models. But as shown in Table 3.8, the AL_4 score is considerably better than any of these 7 models. This example is shown in Figure 3.13, and is a good illustration of how the protein recombination algorithm works. In this case the recombinant model includes two large fragments, from two models built from different templates, obtaining a final composite model that can be better equivalenced to the experimental (in AL_4 terms). The algorithm took the better sections from each of the two models to build an improved, hybrid, model. Figure 3.13B shows the set of possible crossover points between these two initial models (marked as *). This limited distribution of points could indicate an important limitation of this technique: useful crossovers between models are only possible if they can be reasonably superimposed in space keeping together fragments with the same sequence.

3.7.6 T0170 (FF domain of HYPA/FBP11, human)

The FF domain is a 60 amino acid residue phosphopeptide-binding motif. Confident template(s) could not be found using our standard sequence similarity tools (this was indeed a FR/NF target, now PDB structure 1H40 (Allen *et al.*, 2002)). Thus we decided to take all ten models provided by Pmodeller and recombine them. Post-CASP analysis shows that the best initial model, based on the homeodomain 1LFB and aligned by 3D-PSSM (Kelley *et al.*, 2000), is much better than the final recombinant model, suggesting that the algorithm tested may not perform very well with small helical proteins. However, repeating the recombination with the post-CASP version of *insilico*PR, which calculates energies/residue, allowing comparison of proteins of different length, provides a recombinant model scoring 58 AL_4 and 46.7 GDT_TS, comparable to the best initial model.

3.7.7 CASP5 overview and analysis for CM targets

CM targets were considered by the assessors the easiest in the experiment, since finding templates for them was trivial. Nevertheless, only five CASP5 targets had more than 40% sequence identity to the optimal templates available at that time in the PDB. In strict terms, these are not easy comparative models (see Figure 1.9), although their alignments are expected to be easier than those in the FR categories. We were also interested in comparing the ability of the algorithm to produce recombinant models for CM targets, indeed this was the initial motivation for this work. Despite the simplicity of the potential energy function, in most cases, the algorithm presented here selected the best possible alignments and templates from the initial available ensemble. In some cases, our recombinant models

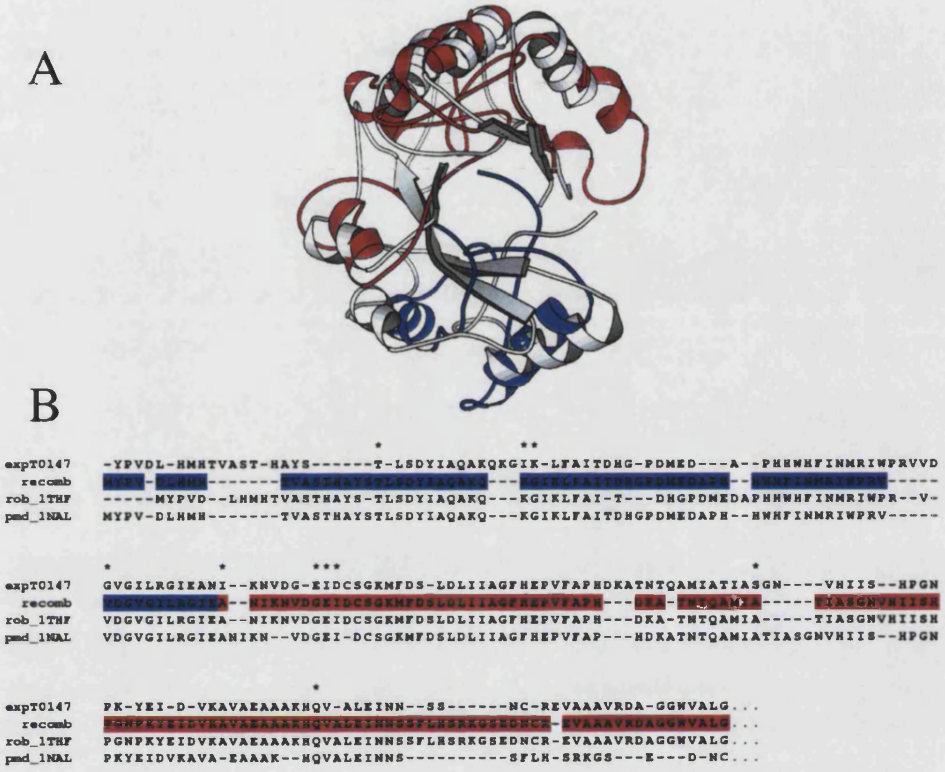


Figure 3.13: (A) Cartoon showing the superposition of the experimental structure for the first 190 residues of T0147 (white) and the recombinant model submitted by our group (N-terminus in blue, C-terminus in red). (B) Corresponding structural alignment of the experimental structure of T0147, the recombinant model and two template-based models generated by Robetta (rob_1THF) and Pmodeller (pmd_1NAL). The N-terminus of the recombinant model was taken from pmd_1NAL, based on that PDB template and highlighted in blue, whilst the C-terminus is derived from rob_1THF, in red. In terms of AL_4 score, the recombinant model is significantly better than both pmd_1NAL and rob_1THF. In terms of GDT_TS, the recombinant model is comparable to them (see text for scores definitions). Asterisks (*) mark possible crossover points between the two initial models after a sequence-based superposition, all within loop regions. The PDB code for the experimental structure is 1M65 (Teplyakov *et al.*, 2003). Three Histidine residues in the recombinant model align to the equivalent Histidines in the experimental structure, the other five are misaligned.

were significantly worse than those constructed by the best predictors. Analysis of some of these results (targets T0137,T0153,T0177,T0178,T0182 and T0192) shows that the quality of the initial models used in the recombination experiments to be the main reason. Particularly, we believe that loop conformations where not successfully sampled for each

initial model. We also noticed that recombination can sometimes improve alignments but at the cost of making GDT_TS scores worse, possibly due to accumulation of errors during the evolutionary procedure. Overall, we have no reason to believe that this recombination procedure works better for FR than for CM targets, although it is expected that alignment errors would be more common in the former. Indeed, an automated ranking per category produced by Michael Levitt (Stanford University), based on the official GDT_TS scores placed our procedure in positions 19 and 20 for both categories, suggesting that the relative performances are similar.

3.8 Molecular dynamics simulations on four CASP5 targets

As mentioned in Section 3.7.1, where we explained our protocol for CASP5, molecular dynamics simulations were done, to assess to what extent they could improve some of our CASP5 predictions, inspired by the work of Lee *et al.* (2001). This part of the work was done in close collaboration with Graham R.Smith, who shared knowledge, tricks and programming code with me. Because of the strict time limitations during CASP and our limited computing resources at the time, only four cases were refined: T0134, T0165, T0177 and T0185, shown in Table 3.10.

3.8.1 Protocol

We used version 3.1.4 of the software package Gromacs (Lindahl *et al.*, 2001) and the OPLS-AA force field (Damm *et al.*, 1997), with its collection of parameters and potential functions. As depicted in Figure 3.14, the input for the procedure is a PDB file, in our case a recombinant model obtained as explained in the previous Chapter. The first step consists of creating a GROMACS topology for all the atoms, including hydrogens, in the PDB-formatted molecule, which describes all the atomic interactions. Next, a cuboid-shaped simulation box needs to be enlarged to accommodate the solvent molecules to solvate our model, using the program *editconf*. The size of the box is increased at least 10Å beyond the longest dimension of the protein. This works well when a periodic boundary is to be used. Then, the box is filled with water molecules, to get a concentration of about 55.5M. In the next step, Na^+ *and* Cl^- ions are added to at least a 0.1M solution, until the total charge is zero. Now the neutralised system is minimised using a steepest descent algorithm, and two equilibration rounds, one for hydrogen atoms and the other for all.

target	comment	run-time
T0134	(FR(H) target) is a delta-adaptin appendage domain, part of a complex (not clathrin-associated) related to lysosome trafficking. It was selected because it has two clear structural subdomains and their relative orientation could change respect to the templates used to model it (1QTS, Ap-2 Clathrin Adaptor α-appendage and 1E42, β-adaptin appendage from clathrin adaptor Ap2).	0.58ns
T0165	cephalosporin C deacetylase (PDB 1L7A) for which several templates in the PDB were found, all of them related to antibiotic biosynthesis. This was considered a CM target, with sequence identities to relative to the templates of around 15%. It was selected for MD analysis for its long loops and to check the packing of a set of α-helices.	0.5ns
T0177	another CM target, a hypothetical protein HP0162 from *H.pylori*, now PDB 1MW7. It was modelled using two bacterial templates (1LFP and 1KON) about 30% identical in sequence. The reason to run molecular dynamics on it was again the subdomain orientation difficulty.	0.74ns
T0185	a CM target from *T.maritima*, eventually annotated as UDP-N-Acetylmuramate-Alanine Ligase (PDB 1J6U), was again chosen because of its complicated arrangement of subdomains. It was modelled using templates of around 25% sequence identity, proteins involved in the biosynthesis of peptidoglycans, therefore, functionally related.	0.5ns

Table 3.10: CASP5 targets selected for molecular dynamics simulations.

The purpose of these is, in theory, to enable the system to reach equilibrium when the subsequent molecular dynamics simulation is performed (Leach, 2001). After all these steps have been accomplished, the molecular dynamics simulations are triggered taking a pair of nodes of a Linux PC farm (866MHz). In all four cases, the length of the step was 0.002ps. The total time simulated is shown on Table 3.10, taking about two weeks. Post-analysis of these simulations consisted of clustering snapshots of the trajectory (one step every four) in terms of backbone RMSD, using the GROMACS tools *trjconv* and *g_cluster*. One conformation from the most populated cluster was then selected and min-

imised, using CHARMM22, and submitted to CASP5 as our refined model.

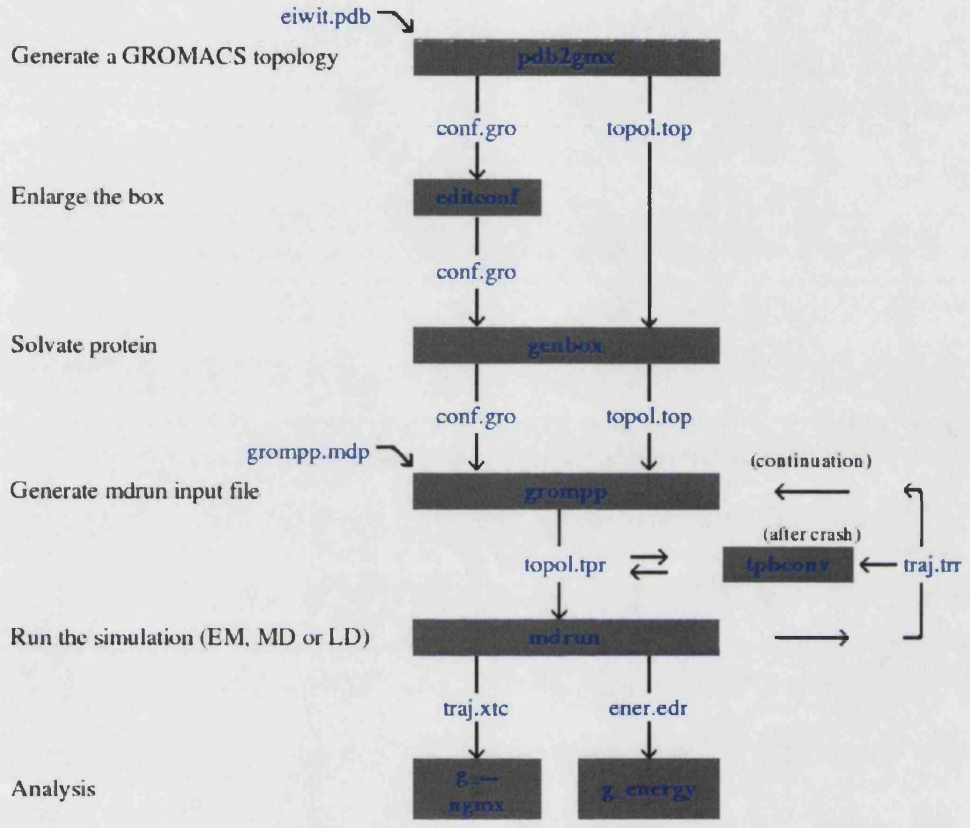


Figure 3.14: Flowchart of GROMACS (taken from http://www.gromacs.org/documentation).

3.8.2 Analysis of results

In the next Sections we compare the MD-refined models submitted to CASP5 to the unrefined models aiming to dissect the effect, negative or positive, that these simulations had on our performance.

T0134 FR(H)

This protein was divided into two subdomains by the CASP5 assessors, T0134_1 and T0134_2, to simplify evaluation. For T0134_1 clearly MD did not improve the model. Upon refinement, GDT_TS moved from 38.78 to 34.84, while AL_4 also deteriorated,

from 74.02 to 59.84. In the case of T0134_2, 0.58ns of MD changed the protein considerably, diminishing the GDT_TS from score from 63.44 to 46.22 and AL_4 from 82.08 to 65.09. So it can be said that MD and subsequent CHARMM22 energy optimisation had a negative impact on the accuracy of our predictions.

T0165, T0177 and T0185 (CM targets)

For these CM targets, Molecular Dynamics simulations also have negative effects over the quality of our models, with final GDT_TS and AL_4 values very close, but generally worse, than the unrefined model, listed in Table 3.11.

target	$GDT\_TS_{unrefined}$	$GDT\_TS_{refined}$	$AL\_4_{unrefined}$	$AL\_4_{refined}$
T0165	49.84	42.53	67.92	58.49
T0177_1	92.98	90.35	100.00	100.00
T0177_2	86.36	84.37	100.00	97.73
T0185_1	63.12	56.43	91.09	87.13
T0185_2	65.73	66.62	86.80	85.28
T0185_3	57.30	52.88	76.92	69.23

Table 3.11: Relative performance of our MD simulations on three CM CASP5 targets as published by the CASP5 assessment. Overall, the use of MD during refinement made a negative impact on GDT_TS and AL_4 scores.

Overall comment on the molecular dynamics refinements

The exact reason for MD not improving the models is not clear, but the fact that energy functions are not perfect is well documented (Lee *et al.*, 2001). Moreover it is not clear which conformations to take after clustering the trajectories as they all seem to have similar energies. Since we only submitted one conformation per each (CASP5 only scored one model per group) our choice of the best model from the ensemble was subjective. If CASP organisers were able to somehow score an ensemble of models, MD might be more objectively analysed in future experiments.

3.9 Conclusions

The genetic algorithm tested in CASP5 tends to produce recombinant models that are comparable to the best initial model, had we identified it, as we also observed in our

in-house benchmark. In addition, this procedure performed well (our group was ranked among the world top 20) in both comparative modelling and fold recognition targets. These results suggests that our simple fitness function correctly identifies good models, making it a good candidate to filter and rank models from automatic servers as well as models built in-house, or indeed a combination of both. In addition, the method has been shown to be able to improve alignments by recombining well aligned regions from individual models. A related methodology presented in CASP5 by Fischer (2003) also obtains good results by executing "cut and paste" operations over protein models, suggesting that this principle is useful, regardless of the implementation. Unfortunately, the quality of the models used to seed the first generation seems to be the upper limit for the quality of the final model, showing that the current implementation of the algorithm is not adding much beyond this baseline. Finally, because good global superpositions are required for useful crossover, the current implementation of *in silico* protein recombination cannot recombine efficiently proteins that are totally different or have different domain orientations. This suggests that local superpositions may be required.

3.10 Possible developments of the recombination methods

To perhaps improve the performance of these recombinant methods several changes or additions to the current algorithms could be done. They are listed here:

- After a crossover event two recombinant proteins can be generated, although at the moment only one of them is being carried over. This could be easily modified in the current code.
- As mentioned earlier, local superimposition of partners in the population could help to generate useful variants, particularly when different folds are being fed into the founder population or if multidomain proteins are used.
- Different mechanisms for generation of structural mutants should be tried since the one tested here is too coarse. Methods such as ϕ/ψ random exploration of hinge regions between secondary structure elements, to generate different packing angles between them, are envisaged. Predicted secondary structure could also guide the building of mutant conformations. Unfortunately, the generation of genuine folding variability will probably require finer energy functions, which leads to the next point.

- A recent paper by Keasar & Levitt (2003) suggests that explicit physical and knowledge-based hydrogen bonding terms in potential functions are very important to help in distinguishing global and local minima in energy landscapes. These terms could be added to our fitness function.

- Non-energetic constraints, used in the the *ab initio* field for the simulation of folding could be added. In particular, clustering of conformations to select the most populated and contact order (average sequence separation between contacting residues) could be used to drive the artificial evolution process on a set of models (Simons *et al.*, 1997b, 1999).

- Inspired by recent work (Zagrovic *et al.*, 2002a,b), conformations close to the average of the population could be calculated by a residue distance matrix, to give extra fitness. In this work they actually calculate atomic distance constraints (calculated by molecular dynamics simulations) and match them to the original nuclear dipoles couplings (Overhauser effect) found in NMR experiments. This could also be attempted here.

- An apparently obvious way to improve the performance of the method would be to energy-minimise protein geometries after recombination events, to relax possible steric clashes added that could marginalise otherwise good models. This has now been partially tested, for a slightly different purpose (see Chapter 4).

- A different interesting approach would be to add functional restraints to the fitness function, if the protein that is being evolved is known to bind particular substrates or partners in well defined ways. The idea could even be used to evolve novel proteins, perhaps evolving their sequence as well, to do pre-designed tasks. Preliminary work on this direction has been recently published by Petersen & Taylor (2003), in which they evolve zinc-binding proteins from scratch.

- An interesting question that remains to be answered in this work is how much each term in the fitness used used, for instance the one presented here, account for the change in alignment shift and RMSD during protein recombination simulations.

3.11 Materials and Methods

Protein test sets from SCOP For every experiment in this paper, protein families from SCOP 1.55 were randomly selected from the 4 major classes (337α, 276β, 374α/β and

391$\alpha+\beta$ families). Only a non-redundant fraction (90% sequence identity cut-off) of protein domains in each family, according to the ASTRAL database (Brenner *et al.*, 2000), was considered. To benchmark *in silico* protein recombination using the simple fitness function, the following SCOP domains were selected as query proteins to be modelled using proteins in the same family as templates (27α,38β,26α/β and 39$\alpha+\beta$, the number of templates used in each case is indicated in brackets): d1pbk_(4), d1pama2(7), d1pne_(6), d1poxa2(3), d2phia_(16), d1pina2(4), d1pvxa_(6), d1pvaa_(5), d1psra_(6), d1ppn_(13), d1a75a(5), d1a5da2(9), d1a25a(5), d1a33_(6), d1a03a(6), d1a0aa(4), d1a0ca(3), d1a1s_1(4), d1a81a1(15), d1ad3a(2), d1adwa(9), d1ae7_(16), d11bga(8), d2abl_2(14), d2act_(13), d1acz_(7), d1qaua(6), d2aaib2(8), d2aaib1(7), d1an8_2(6), d1an4a(4), d1qnna2(10), d1qnga(8), d1qo8a3(3), d1aoza3(2), d1aoa_2(3), d1aoga1(7), d1alo_3(5), d1allb(11), d1ala_(9), d1qlca(6), d1qk1a1(4), d1qkka(9), d1qh7a(6), d1aisa2(7), d1aisa1(5), d1ain_(9), d1aw0_(5), d1aw1a(8), d1awpa(3), d1awca(4), d1qpca(5), d2apr_(11), d1qqya(8), d1qqka(5), d2ay1a(6), d1ayaa(16), d1b26a2(2), d1b2pa(5), d1b06a2(10), d1b1xa1(8), d1b8za(3), d1bg3a3(3), d1bg0_1(5), d1be9a(4), d2bb2_1(9), d1bb9_(15), d1rbla2(5), d1rblm(5), d1bc4_(9), d1blxb(4), d1bla_(5), d1bjwa(6), d1bkja(3), d1bkb_2(2), d1bh6a(6), d1bhda(3), d1bwva2(5), d1bwya(13), d8ruci(5), d1burs(5), d2rspa(4), d1rp1_2(5), d1bzsa(8), d1bxta2(6), d1bxsa(2), d1c4zd(4), d1c1da1(2), d1c9ha(4), d1cf5a(6), d1ce7a(6), d1scha(4), d1clh_(8), d1ck7a2(8), d1sw6a(4), d2ctha(6), d1ste_1(3), d1crka1(5), d1srra(9), d1crb_(13), d1cs8a(14), d1csee(6), d1cpcb(12), d1cpn_(2), d1cpt_(3), d1cyda(3), d1d6aa(7), d1d3ca2(7), d8dfr_(3), d1teha1(6), d1tcda(8), d1dn2a2(14), d1tnra(3), d1dot_1(7), d1dlpa2(6), d1dmxa(3), d1dt0a1(8), d1duvg2(4), d1duxc(5), d2trxa(7), d1dssg2(5), d1dsya(5), d1tx4b(11), d1e3pa2(2), d1e3ia1(6), d1e1oa1(2), d1u9aa(4), d1ef5a(3), d1egza(4).

For the detailed analysis presented only eight SCOP families were considered, two from each class. Each contained several templates with a variable degree of sequence identity to the query. They were: d1a03a (rabbit calcyclin, 1A03), d1a8h_1 (*Thermus thermophilus* methyonil-tRNA synthetase, 1A8H), d1qfja1 (*Escherichia coli* flavin oxidoreductase, 1QFJ), d2phla1 (*Phaseolus vulgaris* seed storage protein, 2PHL), d1pmt_2 (*Proteus mirabilis* glutathione transferase, 1PMT), d1poxa2 (*Lactobacillus plantarum* pyruvate oxidase, 1POX), d1pne_ (bovine profilin, 1PNE) and d1a5r_ (human small ubiquitin-related protein SUMO-1, 1A5R).

Single vs. Multiple-template modelling 271 families from SCOP were randomly selected. A draw was made to select one protein domain (query) in each family to be modelled using the other proteins, in the same family, as templates. Templates in each family were ranked on sequence identity to the query. Only the first was used for single-template models and the first five for multiple template models. To bypass alignment errors in this

experiment, the query sequence was aligned to the best template using the known molecular structure (taken from the PDB). The query and the best template were structurally aligned and superimposed in space using *msuper*. In our implementation, two given Cβ are considered to be equivalent if their distance is less than 3Å. When more than one template was used, a multiple structural alignment was built and only the leader sequence was then aligned to the query. The program 3D-JIGSAW builds multiple-template models by a combination of mean-field selection of superimposed fragments and side-chain optimisation (Bates & Sternberg, 1999).

Optimal and suboptimal sequence alignments When query and template sequences needed to be aligned, we used *Profile1*, using pssms computed after 5 iterations of PSI-BLAST against the nr database (http://www.ncbi.nlm.nih.gov) with default parameters. After computing the optimal alignment, the pssm is used to calculate the average log-odd score (or bit-score) per residue. Alignments were only considered for the experiments if their bit-score was over 2.0. To generate suboptimal alignments, the guidelines explained in detail in previous papers (Saqi & Sternberg, 1991; Saqi *et al.*, 1992) were followed to implement an iterative dynamic programming function that discovers non-trivial suboptimal alignments by penalising positions aligned in previous iterations. After computing one alignment trace, aligned residues are marked to be penalised in the next iteration. The penalty chosen for next iterations was -0.1.

Atomic deviation measures For the experiments presented in Sections 3.1,3.2,3.3 reported RMSD values were obtained after superimposing pairs of models with the program SSAP. These measures correspond to average deviations between all pairs of equivalent Cα. For the recombination experiments, the RMSD calculations are now based on Cβ and are calculated as part of *msuper*. Both measures are based on all the equivalent pairs of residues obtained after aligning two sequences, including loops.

Computing time A recombination experiment can take from 5 minutes to several hours (running serial C++ code on a 2.4GHz Pentium IV desktop PC under Linux) depending on the size of the sequence to model and the population. Thus it is usually more expensive than building models using traditional methodologies. The most time-consuming step of the algorithm is growing each population, but this could be done in parallel if a farm of computers is available by performing one reproduction event per node.

Alignment shift calculation To calculate the quality of the alignments in the protein recombination experiment, the resulting models in each population were structurally aligned to their corresponding experimental structures, as taken from the PDB database. Taking these alignments as references, the average number of shifts per aligned residue is computed. As models and real structures have identical sequences this computation is trivial. An average shift of 0 means that the real structure and the model can be optimally superimposed using their corresponding sequence alignment. A value of 1 would mean that every residue is displaced, on average, one residue.

Generation of models from shifted alignments The sequence for each of the eight query SCOP domains (described above) was used as input for the interactive form of the web server 3D-JIGSAW (see http://www.bmm.icnet.uk/servers/3djigsaw) and 5 alignments to the top template (100% identical in sequence) were shifted 1,2,3 or 4 positions to either side of a randomly selected residue before building the models. The resulting complete models were used in the recombination experiment.

Building models from PSI-BLAST alignments PSI-BLAST version 2.2.5 was used with default parameters. The database used was dPFAM_PDB, the same one used by our 3D-JIGSAW server. Five iterations were used and the output was parsed to extract the alignments to a maximum of 8 templates. Models were built from these alignments using 3D-JIGSAW. The average e-value of the alignments used was $8e^{-3}$. PSI-BLAST models were on average 1.7 residues shorter than corresponding models aligned by our procedure.

Chapter 4

Exonic structure and recombination of proteins domains

As introduced in Section 1.1.5, introns are fragments of non-coding DNA intercalating gene-coding regions (exons). In general, the proportion of non-coding regions in eukaryotic genes, including introns, is higher than that of coding regions. Since genetic recombination normally relies on a random crossover point being drawn along a stretch of two homologous DNA strands pairing together, the more non-coding sequences existing in a gene, the less chances are for this crossover point to be inside a coding region. Since eukaryotic genes may have large introns, it can be postulated that they have a role in mediating genetic recombination.

Introns can be classified according to different criteria, including splicing mechanism, sequence signals or even their late/early origin. However, from a phylogenetic point of view, assuming that intron gain or removal are rare events, introns are considered to be homologous regardless of their type, sequence or length, as long as they occupy homologous positions in the DNA (Patthy, 1999). They will be considered in this way in this chapter, in the context of protein structure. There is one informative attribute of introns that can be easily calculated from the genomic data, the phase. The phase relates the position of consecutive introns to the final spliced reading frame. Since RNA is translated in triplets (codons), the phase of introns can only be 0, 1 or 2. Phases are important since changing them may radically change how a mRNA moleculae is read and translated.

Intron gain is frequently associated with insertion of mobile genetic elements whilst intron loss or shortening is often explained by a model in which homologous recombination between the genomic copy of a gene and an intron-less DNA produced by reverse transcription of the corresponding mRNA eliminates the genomic intron (Patthy, 1999;

Mourier & Jeffares, 2003). Intron sliding can also occur, although it has been reported as an infrequent event (Stoltzfus *et al.*, 1997). Regardless of their origin, introns must be spliced from their mRNAs for proteins to be translated. Intron splicing relies on very short RNA motifs marking the boundaries; changes in these positions, which are the only ones conserved along introns, will directly affect the outcome of the splicing process (Padgett *et al.*, 1986; Alberts *et al.*, 1994; Clark & Thanaraj, 2002). For this reason, introns are potential places for insertion or deletion of fragments in proteins, thus potential places for significant changes in protein structure. Several studies reviewed by Patthy (1999) illustrate how transposable introns can potentially modify protein structure by adding or removing small peptides inside the host fold. Insertion of this sort of introns is more likely to be selectively neutral if most of the transposon is removed upon mRNA splicing; introns would then be evolutionarily accepted in positions of the fold that can tolerate the insertion or deletion (if an imperfect excision occurs) of a few residues.

Introns can be located separating complete functional domains, as more traditionally thought of in terms of protein evolution (Chothia *et al.*, 2003), but they can also split the exonic components of individual functional domains. We will further explore this here, trying to investigate whether intron-exon boundary (IEB) information could potentially be useful in protein design and modelling.

It is generally accepted that rational protein design involves searching vast sequence and conformational spaces (see for instance Looger & Hellinga (2001)). To reduce the search space, many of the early design attempts have focused only on redesigning proteins with a fixed backbone, or allowing small backbone movements (Reina *et al.*, 2002; Looger *et al.*, 2003). If significant modifications of functions are to be accomplished, perhaps larger backbone movements will be needed. However, to accomplish this it is necessary to know how to accommodate these large changes whilst keeping the protein fold stable. A possible approach could be using an ensemble of homologous proteins and identifying key hybridisation points. Indeed recent work in this direction has been conducted experimentally (Voigt *et al.*, 2002). Following this lead, which suggests that IEBs could lie at special locations within protein folds, we considered two completed eukaryotic genomes, mouse and man, and decided to look at the protein structure level to check if IEBs are indeed different to the other residues in a protein.

4.1 Intron survey within protein structures

This part of the work was done in close collaboration with Pall F.Jonsson, graduate student in the Biomolecular Modelling Laboratory, and therefore he is acknowledged here as co-

author of the results shown within this section (Contreras-Moreira *et al.*, 2003c).

A set of 684 human and mouse protein structures, and their amino acid sequences, extracted from the PDB (see Section 4.7) was taken as the sample for the following statistical analysis. In this Section, IEB residues are defined as those sitting immediately to the left of a given intron at the DNA level.

4.1.1 Secondary structure context of IEBs

Residues at IEBs were assigned a secondary structure type as calculated by the program DSSP. A simple analysis was done to compare the secondary structure nature at the boundaries to the expected (background) frequency of secondary structure states on the same dataset. The results, shown in Table 4.1, show a significant preference for intron boundaries to occur in coil regions of proteins and less inside α-helices and extended β-strand elements. This could indicate that insertion of introns into sections of ordered structure, such as α-helices and β-sheets, is likely to affect the overall structure and function which, in return, affects the protein's fitness in natural selection terms. Furthermore, even when boundaries occur within strands and helices, they tend to be close to the end of their secondary structure element, as shown in Table 4.2. This is especially apparent for exon boundaries in extended strands. The data supports the hypothesis that boundaries tend to occur in less ordered areas. However, as shown in Table 4.1, IEB residues seem to have no overall preference with respect to the phase of their adjacent exons. This fact does not necessarily contradict previous models predicting that some phase arrangements are preferred within the same gene or between homologous genes to allow intronic recombination (Patthy, 1999).

After this survey was done we found in the literature similar observations, in agreement with our data, although extracted from very small datasets (Craik *et al.*, 1982, 1983).

4.1.2 Local structural variability at IEBs

The relationship between structure conservation and IEBs was studied by mapping the boundaries on pairs of homologous human and mouse PDB structures with a pairwise sequence identity $\geq 40\%$. These pairs were structurally aligned (using *msuper*) and structural deviations at boundary positions compared to the overall deviation between each (see Section 4.7 for details). The structure conservation of boundaries in coil regions, helices and strands was not found to differ significantly from the expected values, as shown in Table 4.3. The location of boundaries does not therefore appear to be in significantly more divergent regions between homologous proteins. Hence, the reason why these boundaries

(3-state) DSSP structure	$freq_{obs}$	$freq_{exp}$	Δ	phase0	phase1	phase2
C - No secondary structure	776	544	**+43%**	279	262	235
C - Isolated β-bridge	29	31	-6%	10	9	10
C - Hydrogen bonded turn	308	288	+7%	106	111	91
C - Bend	260	265	-2%	90	72	98
E - extended β-strand	430	537	**-20%**	130	148	152
H - α-helix	570	702	**-19%**	199	174	197
H - 3_{10}-helix	73	80	-9%	27	22	24
H - 5-helix	1	0		0	1	0

Table 4.1: Observed and expected frequencies of IEB within DSSP assigned secondary structure elements. The total number of intron residues is 2447, out of a total of 116,740 residues. The most significant differences are highlighted in bold. The observed differences between the observed frequencies and the expected according to the background are highly unlikely to be random, according to a χ^2 test with 7 degrees of freedom ($p \ll 0.001$). The three right columns show the phase of the preceding exon for each IEB. We found no overall differential distribution of IEBs with respect to exon phases.

set of IEBs	end_{obs}	end_{exp}	mid_{obs}	mid_{exp}	$p(\chi_1^2)$
all β-strands	184 (0.41)	45 (0.1)	266 (0.59)	405 (0.9)	$p = 9.3 \cdot 10^{-106}$
conserved β-strands	13 (0.21)	6.2 (0.1)	49 (0.79)	55.8 (0.9)	$p = 0.004$
all α-helices	114 (0.2)	57.9 (0.1)	465 (0.8)	521.1 (0.9)	$p = 7.7 \cdot 10^{-15}$
conserved α-helices	15 (0.25)	6 (0.1)	45 (0.75)	54 (0.9)	$p = 0.0001$

Table 4.2: Frequency of intron-exon boundaries appearing at the ends of extended β-strands and α-helical structures. Ends are defined as the first or last 5% of the secondary structure element length. Shown are the frequencies for all exons as well as the subset of conserved exons between mouse and human. The differences are significant according to χ^2 tests with 1 degree of freedom.

are preferentially found in coils and at the ends of α-helices and β-strands is not clear. Perhaps this is to allow variable packing of exons. To assess this we compared the packing of exons in homologous proteins.

4.1.3 Packing of exons using structural alignments

We used a method based on *msuper* alignments to assess whether exons can have alternative packing arrangements with hinge points located on IEBs. For this study the previously described set of homologous human-mouse sequence pairs was used. Each pair was aligned by sequence and two adjacent windows, representing two exons of aver-

bin (σ)	C_{obs}	C_{exp}	H_{obs}	H_{exp}	E_{obs}	E_{exp}
-1.5	1	1	1	2	0	1
-1	4	7	13	9	9	9
-0.5	85	90	88	86	37	58
0	170	139	73	75	77	67
0.5	40	46	18	17	14	15
1	24	32	8	9	10	8
1.5	17	18	6	5	5	5
2	14	15	1	4	4	2
2.5	8	9	1	2	1	1
3	2	6	0	1	1	1
3.5	2	3	0	1	5	1
4	2	3	1	0	5	0

Table 4.3: Structural conservation of IEB residues after structural superimposition. Observed and expected values are shown for coil (C), extended β-strands (E) and α-helices (H) after standardising the original data (in the range [0-9] Å). These distributions are not significantly different according to a χ^2_{11} distribution, with p values over 0.2.

age length, were shifted along the sequence pairs, and a structural alignment performed by superimposing the two left hand exons on each other, carrying over the structure of the right hand exons, as described in Section 4.7. Flexibility at each position was assessed as the angle between vectors from the N-terminus to the centre of geometry of each of the right hand exons (see insert to Figure 4.1). This angle was used as an indication of the structural deviation between the pair at each point. No significant difference ($p = 0.62$ for a χ^2 test, with 12 degrees of freedom) was found in the distribution of angles at IEBs compared to background distribution as shown in Figure 4.1. This would suggest either that evolution does not favour increased diversity of packing between homologous exons or the method we used is not sensitive enough to pick up hinge points in boundary locations.

4.1.4 Analysis of tertiary structure contacts

Previous work suggests the importance of tertiary contacts in understanding the interactions between components of a fold (Voigt *et al.*, 2002; Berezovsky *et al.*, 2000; Berezovsky & Trifonov, 2001). Trying to understand our findings, we also looked at the distri-

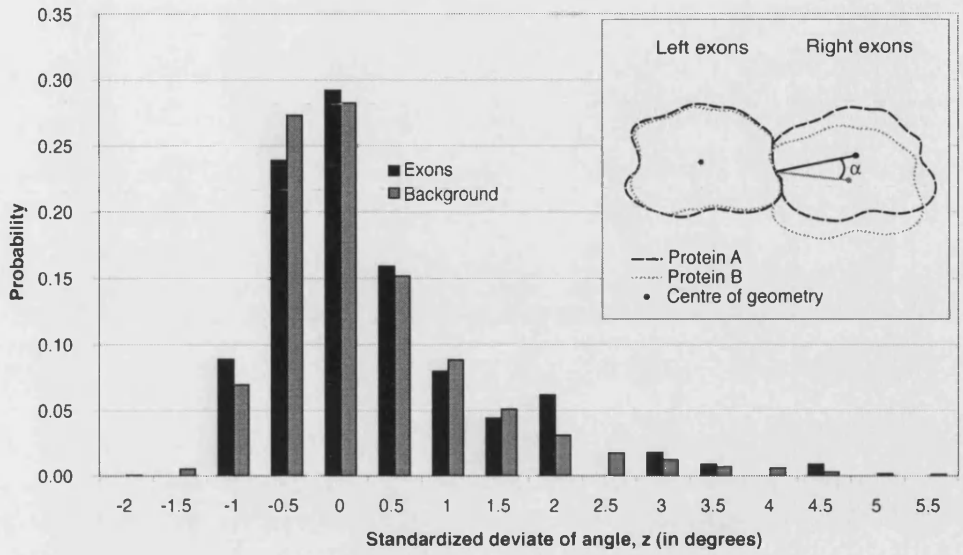


Figure 4.1: Distribution of standardised normal deviates of angles in intron-exon boundaries (black) and the background (grey) with a mean value of 8.1 degrees and standard deviation of 6.7 degrees. Greater Z values represent higher degree of variability between a homologous pair at a specific position. There is not a significant difference between the samples ($p = 0.62$ for χ^2_{11}). The insert shows a schematic diagram of the calculation on a pair of proteins consisting of two exons. Centres of geometry are depicted. By superimposing the left-hand exons and carrying over the right-hand exons as rigid bodies, an angle α can be measured (Figure courtesy of Pall F.Jonsson).

bution of contacts around IEBs as compared to non-boundary residues along the primary sequence. Much work has been done in the past to address the conservation of introns by building multiple alignments of homologous sequences from different organisms (Fedorov *et al.*, 2001, 2002; Betts *et al.*, 2001). Despite the limitation of using only human and murine proteins, we also wanted to check if conserved and non-conserved introns are different in terms of contacts. Results (Figure 4.2 A) show that, in our relatively large dataset, boundary-residues are in general no different, in terms of their contact profile, compared with the rest of the protein. Low-contact regions are preferably occupied by coil residues, irrespective of the existence of a boundary there. However, as shown in Figure 4.2 B and C, coil boundary-residues seem to be preferred for low-contact regions in the subset of conserved boundaries.

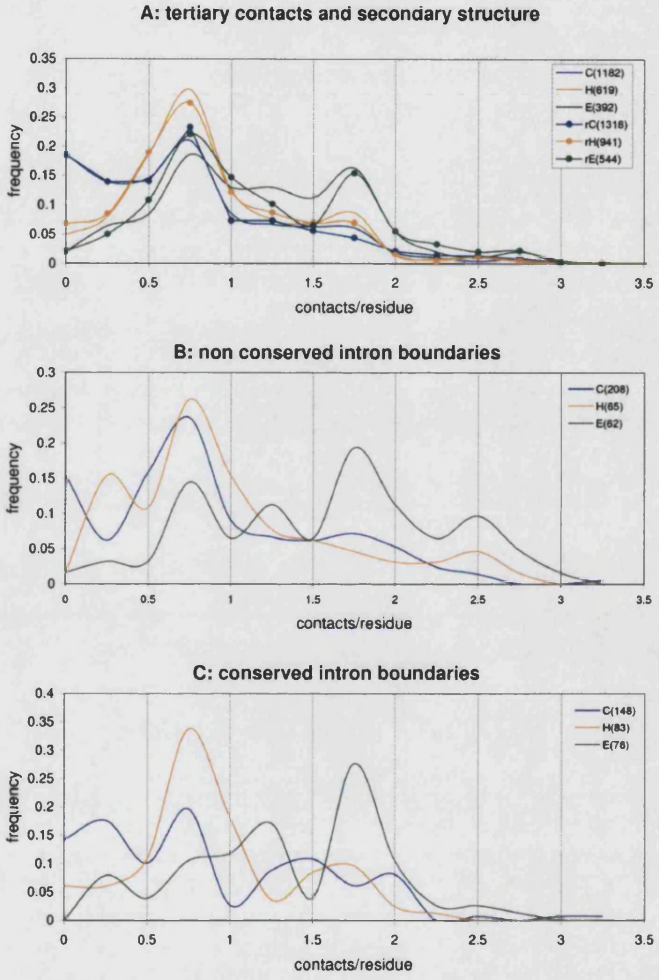


Figure 4.2: (A) Distribution of contacts per residue in a population of intron boundaries as compared to a population of randomly chosen residues. Contacts are calculated as explained in Materials and Methods, by checking residues to the right of the selected position (intron or randomly selected) of the sequence with residues to the left. The original distribution of contacts along each sequence is smoothed by averaging with a window of size 5. Three different distributions are plotted, according to the 3-state secondary structure of the selected position, where C corresponds to coil conformations, H to helices and E to extended strands, as stated in Table 4.1. Random residues are labelled rC, rH and rE. The number of observations is shown in brackets. (B) and (C) Distribution of contacts for non-conserved and conserved intron boundaries for a set of non-redundant homologous pairs of human and murine proteins. These distributions were smoothed as explained above, with a window of size 5.

4.1.5 Location of IEBs in relation to functional sites

Details of functional residues of proteins in our dataset were extracted from the PDB and the spatial relationship between exons and functional sites examined. A total of 94 functional sites (as defined in PDB 'SITE' records) were obtained from 68 PDB structures (listed in Table 4.4). From the total of 308 IEBs contained in this subset, 18% (55/308) are located in the vicinity ($distance < 7\AA$) of the functional site. Similar proportions are obtained when the same calculation is repeated on sets of 308 randomly sampled residues, suggesting that on average there is no special preference for IEBs to be near important functional sites.

When examining the exon-composition of functional sites we found that 34% (106/308) of intron boundaries in our set separate residues forming these sites. In total, 48 out of 94 functional sites contain residues belonging to separate exons. Again these observations follow similar proportions as those obtained when repeating the calculations with randomly chosen residues, suggesting that this is not an exclusive feature of intron boundaries. In summary, these results suggest that the pressure of selection that boundary-residues support, in relation to their effect on the protein's function, is not different from that of the rest of the protein.

Table 4.4: Description of the 94 functional sites used in this work, as extracted from the PDB. The 'Residues' column indicates the number of residues within each site.

PDB chain	Site	Residues	PDB Annotation
1cffa	CA1	5	Calmodulin
1cffa	CA2	5	
1cffa	CA3	5	
1cffa	CA4	5	
3ayka	ZNA	3	matrix metalloproteinase
3ayka	ZNB	3	
3ayka	CAB	3	
3ayka	CGS	12	
1gs4a	AC1	11	human androgen receptor, ligandbinding domain (cortisol)
1gs4a	AC2	5	
1rpma	ATE	1	protein tyrosine phosphatase mu
2gmfa	REA	14	human granulocyte macrophage colony stimulating factor

(continued on next page)

Table 4.4:

(continued from previous page)

PDB chain	Site	Residues	PDB Annotation
1gula	GTE	11	glutathione transferase
1gula	HTE	8	
1h4wa	CAT	3	structure of human trypsin IV (brain trypsin)
1h4wa	BEN	8	
1h4wa	CA	4	
1h6fa	MO6	3	tbx3, t-box transcription factor, ulnar-mammary syndrome
1h6ha	AC1	8	px domain from p40phox bound to phosphatidylinositol 3-phosphate
1mema	CAT	3	crystal structure of cathepsin k complexed with a potent vinyl sulfone inhibitor
1vhra	RCA	11	human vh1-related dual-specificity phosphatase
1bio	NUL	3	human complement factor D in complex with isatoic anhydride
1gxca	TPB	5	fha domain from human chk2 kinase in complex with a synthetic phosphopeptide
1h8dh	AC1	14	human alpha-thrombin complex with a tripeptide phosphonate inhibitor
1klt	CIC	3	pmsf-treated human chymase (Serine protease)
1mfma	ZN	5	copper,zinc superoxide dismutase
1mfma	CU	4	
1trna	CAT	3	trypsin (e.c.3.4.21.4) complexed with the inhibitor diisopropyl-fluorophosphofluoridate
1h9oa	PTR	7	phosphatidylinositol 3-kinase, p85-alpha subunit
1kpf	HNE	3	protein kinase pkci-1 with inhibitor
1kpf	AVE	1	
5gdsh	CAT	3	human alpha-thrombin:hirunorm V complex
1bp3a	ZNA	2	growth hormone-prolactin receptor complex
1bsxa	A	9	thyroid hormone receptor beta
1c25	DSU	2	cdc25a catalytic domain
1c25	POP	7	

(continued on next page)

Table 4.4:

(continued from previous page)

PDB chain	Site	Residues	PDB Annotation
1hazb	CAT	3	porcine pancreatic elastase and human beta-casomorphin-7
1qf8a	ZF1	4	casein kinase beta subunit
1buia	ASA	3	microplasmin-staphylokinase complex
1fit	AVE	1	fragile histidine triad protein(chromosomal translocation)
1fj2a	ACA	3	human acyl protein thioesterase
1hd2a	BEZ	10	antioxidant enzyme human peroxiredoxin
1hdoa	AC1	24	biliverdin-ix beta reductase:NADP complex
1qh5a	ZNA	8	human glyoxalase ii with s-(n-hydroxy-n-bromophenylcarbamoyl)glutathion
1hh8a	FLC	10	phagocyte oxidase factor
1znca	CTA	5	human carbonic anhydrase IV(lyase)
2fha	FOX	8	human H chain ferritin
1e42a	AC1	5	beta2-adaptin appendage domain from clathrin adaptor ap2 (Mg)
1qnta	ACC	1	human o6alkylguanine-DNA alkyltransferase
1qr2a	ZNA	3	human quinone reductase type 2
1uch	CAT	4	deubiquitinating enzyme uch-l3(Cysteine protease)
2hft	VII	5	human tissue coagulation factor
2hhma	M1	7	human inositol monophosphatase (e.c.3.1.3.25) complex with gadolinium and sulfatehydrolase
1e9ea	TMP	10	human thymidylate kinase (f105y) complexed with dtmp
1e9ea	ADP	12	
1sra	EF1	5	calcium-binding protein (osteonectin)
1sra	EF2	5	
1sra	MET	3	
1eaxa	SO4	4	matriptase, membrane-type serine protease
1eaxa	BEN	8	

(continued on next page)

Table 4.4:

(continued from previous page)

PDB chain	Site	Residues	PDB Annotation
1eaza	LBS	8	phosphoinositol (3,4)-bisphosphate binding PH domain of tappl
1aoxa	MGA	5	I domain from integrin alpha2-beta1
1ap6a	MNA	4	human mitochondrial manganese superoxide dismutase
1b08a	CR1	5	lung surfactant protein D(sugar binding)
1autc	CAT	3	human activated protein C
1rbp	R1	9	retinol binding protein
1rbp	R2	7	
1ggla	LBS	5	human cellular retinol binding protein III
1pina	ACT	3	pin1 peptidyl-prolyl cis-trans isomerase from Homo sapiens
1gkda	BUA	4	matrix metalloprotease MMP9 active site mutant-inhibitor complex
1gloa	CAT	3	cys25ser mutant of human cathepsin S
1icfa	ACT	2	cathepsin l(Cysteine proteinase)
1ido	MG	6	I-domain from integrin CR3, Mg2+ bound
1cyna	BIN	13	cyclophilin B complexed with [d-(cholinylester)Ser8]-cyclosporin
1gmya	ACT	1	cathepsin B complexed with dipeptidyl nitrile inhibitor
1gnua	NI	2	GABA(A) receptor associated protein gabarap
1o7ka	API	2	human p47 PX domain complex with sulphates
1o7ka	APA	3	
1psra	HO	4	human psoriasin (s100a7),Ca2+ substituted for HO3+ (EF-hand protein)
1rlw	CR1	12	calcium-phospholipid binding domain from cytosolic phospholipase A2
1rlw	CR2	5	
1rlw	CR3	8	
1rlw	CA1	1	

(continued on next page)

Table 4.4:

(continued from previous page)

PDB chain	Site	Residues	PDB Annotation
1rlw	CA2	1	
2mfn	RGD	3	cell attachment modules of mouse fibronectin containing the rgd and synergy regions
2mfn	SGY	5	
1npma	ACA	3	neuropsin, a Serine protease expressed in the limbic system
1vhh	ZN1	4	amino-terminal domain (residues 34 - 195) of signalling protein sonic hedgehog
1eaqa	CL1	3	runx1 runt domain: structural switch and bound chloride ions modulate DNA binding
1ao5a	A	3	mouse glandular kallikrein-13 (prorenin converting enzyme)
1glqa	GA	7	transferase(glutathione)
1glqa	HA	5	
1gmla	AC1	2	mouse CCT gamma apical domain(chaperone)
2znc	ZN	3	murine carbonic anhydrase IV

4.2 *in silico* recombination crossover hot spots seem to avoid IEBs

Taken together, the results presented so far suggest that there is some evolutionary feedback between where introns reside in genes and the proteins coded by those genes, although this might have only weak connections to protein function. In terms of protein evolution, it would make sense to think of introns being placed into the more flexible or loosely packed parts of a fold, because that way the risk of disrupting the protein if the intron boundaries are lost or substantially modified (for example through insertion of a domain), is minimised Patthy (1999). Therefore, it should be possible to find places inside particular protein folds where introns are more likely to occur. Put in a different way, introns could be marking places along a fold primary structure, and the corresponding gene structure, where it is easier to modify proteins while maintaining the fold. However, as

seen in the previous sections, contacts or flexibility alone are not enough to identify these positions. To explore how these boundaries could be located, the following experiment was carried out. A group of 22 human and murine proteins, extracted from the initial PDB dataset, was selected as explained in Section 4.7. For each of them, comparative models were built using as many templates from the same or different species as possible. This included many templates for which we had no information on intron placement and even bacterial proteins with no introns at all. This information is summarised in Table 4.5. The resulting 22 populations of models were subsequently recombined. The recombination protocol was modified (see Section 4.7) in order to allow crossover points to occur in any residue and to improve the mutation mechanism. Results are shown in Figures 4.3 and 4.4. From a total number of 71 boundary-residues found in the dataset, 56 (79%) have less than 5% of frequency of recombination (compared to 65% expected by chance, $p = 0.01$ for χ^2_1). In other words, the observed crossover hot spots in the 22 recombinant populations of proteins tend to occur away from natural IEBs, although this correlation is weak. Hence, we essentially obtain a blurred negative image of IEB location by the use of our synthetic recombination approach. This is likely to be a consequence of the rigid crossover protocol, that is unable to emulate the natural accommodating flexibility of proteins. Since our artificial protein recombination protocol ignores where introns are and only optimizes the structural fitness of a population of proteins, these results suggest that location of introns is an important factor affecting protein fitness, in agreement with genetic evidence (Patthy, 1999). Voigt *et al.* (2002) proposed in a recent paper that the correlation between introns and protein building blocks could occur as a result of natural selection, regardless of their early or late origin. However, as Figures 4.3 and 4.4 show, contact profiles were calculated for each of the 22 populations and no spatial correlation could be seen between regions with relatively few contacts and natural IEBs, as would have been expected. This suggests that it may be too simplistic to assume that boundaries separate autonomous sections within proteins.

Table 4.5: Subset of 22 proteins used in the recombination experiments.

PDB	(PFAM family) annotation	Templates and sequence identity range	Origin of homologous templates
1f5xa	(PF00621) Rho GEF domain	9, 100%-19%	Homo sapiens, Mus musculus
1bc9	(PF01369) Sec7 guanine-nucleotide-exchange factor domain	3, 100%-37%	H.sapiens, Saccharomyces cerevisiae
1bci	(PF00168) C2 domain of cytosolic phospholipase A2	19, 100%-20%	H.sapiens, Rattus norvegicus, Rattus rattus
1a66a	(PF00554) Rel homology domain, eukaryotic transcription factor	11, 100%-23%	H.sapiens, M.musculus, Anopheles gambiae

(continued on next page)

Table 4.5:

(continued from previous page)

PDB	(PFAM family) annotation	Templates and sequence identity range	Origin of homologous templates
1ak6	(PF00241) Cofilin/tropomyosin-type actin-binding protein	9,100%-22%	H.sapiens, M.musculus, Sus scrufa, Acanthamoeba castellanii, S.cerevisiae, A.thaliana
1bv8a	(PF00207) Alpha-2-macroglobulin	3, 100%-62%	H.sapiens, Paracoccus denitrificans, R.norvegicus
1b4qa	(PF00462) Glutaredoxin	10, 100%-20%	H.sapiens, phage T4, E.coli, S.scrufa
1ayk	(PF00413) Matrixin, metalloprotease	15, 100%-59%	H.sapiens, S.scrufa
1cmza	(PF00615) Regulator of G protein signaling domain GAIP	7, 100%-31%	H.sapiens, R.norvegicus, Bos taurus
1gcf	(PF00041) C-terminal domain of granulocyte colony-stimulating factor receptor	10, 100%-16%	M.musculus, Oryctolagus cuniculus, H.sapiens,Ovis aries
1blj	(PF00017) BLK SH2 domain	19, 100%-51%	M.musculus, H.sapiens, Rous's sarcoma virus, Gallus gallus
1ceea	(PF00071) Ras family, CDC42	21, 100%-42%	H.sapiens, M.musculus, Salmonella typhimurium
1etc	(PF00178) Ets-domain	14, 100%-36%	M.musculus, H.sapiens
1df3a	(PF00061) Lipocalin / cytosolic fatty-acid binding	20, 100%-16%	M.musculus, B.taurus,S.scrufa, R.norvegicus
1l3na	(PF00080) Copper/zinc superoxide dismutase	12, 100%-27%	H.sapiens, S.typhimurium, E.coli, Spinacea oleracea,B.taurus, Xenopus laevis,Photobacterium leiognathi,Actinobacillus pleuropneumoniae, S.cerevisiae
1gnc	(PF00489) Interleukin-6/G-CSF/MGF family	10, 100%-15%	H.sapiens, C.familiaris
1iy3a	(PF00062) C-type lysozyme/alpha-lactalbumin family	11, 100%-34%	H.sapiens, Phasianus colchicus, Cavia porcellus, B.taurus, Capra hircus, Tachyglossus aculeatus, Oncorhynchus mykiss, G.gallus, Canis familiaris, Equus caballus, Coturnix coturnix
1gd5a	(PF00787) PX domain	4, 100%-12%	H.sapiens, Staphylococcus aureus, S.cerevisiae
1glqa	(PF00043) Glutathione S-transferase	11, 100%-16%	H.sapiens,Zea mays, M.musculus,A.thaliana
1f16a	(PF00452) Apoptosis regulator proteins, Bcl-2 family	12, 100%-16%	H.sapiens, R.norvegicus, E.coli, M.musculus, Kaposi's sarcome herpesvirus
1ig6a	(PF01388) ARID/BRIGHT DNA binding domain	7, 100%-20%	H.sapiens, Drosophila melanogaster, E.coli, S.cerevisiae
1h4wa	(PF00089) Trypsin	14, 100%-38%	R.rattus, S.scrufa, B.taurus, H.sapiens, E.coli, R.norvegicus

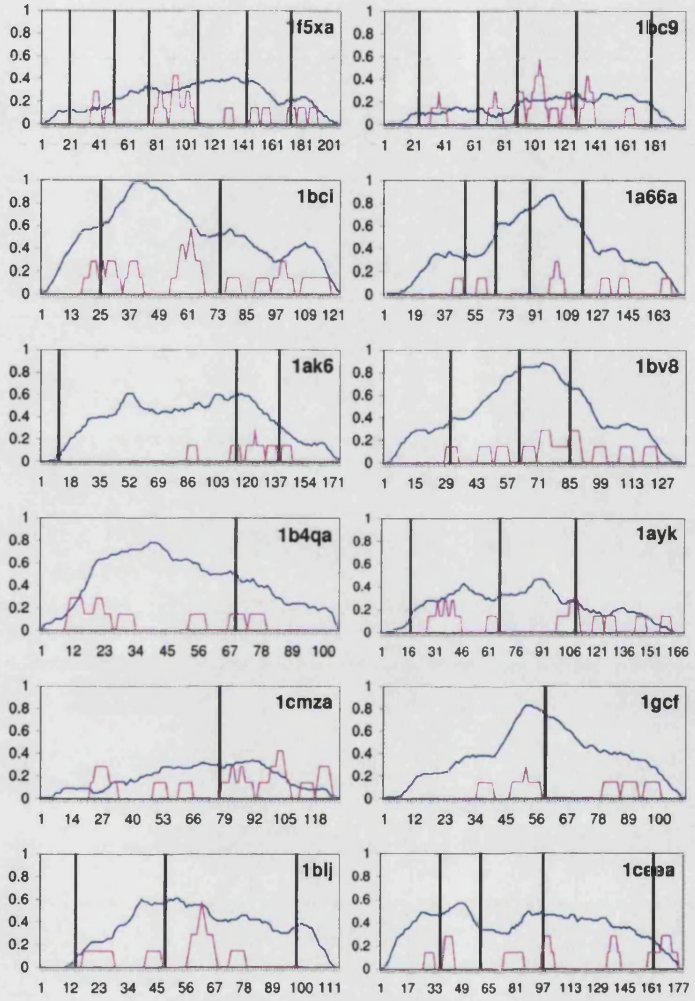


Figure 4.3: Frequency of crossover (pink) and tertiary contacts (blue) along the primary sequence of 12 human and mouse proteins. Vertical bars indicate where natural intron boundaries are found in the human or mouse sampled proteins. Crossover frequencies were smoothed by averaging inside a window of length 7 (similar plots are obtained with other values). The Y-axis shows the observed frequency of crossover in each of the evolving protein populations and the number of contacts divided by the length of the protein. The X-axis represents the amino acid sequence of each protein. Contacts are calculated as explained in Section 4.7.

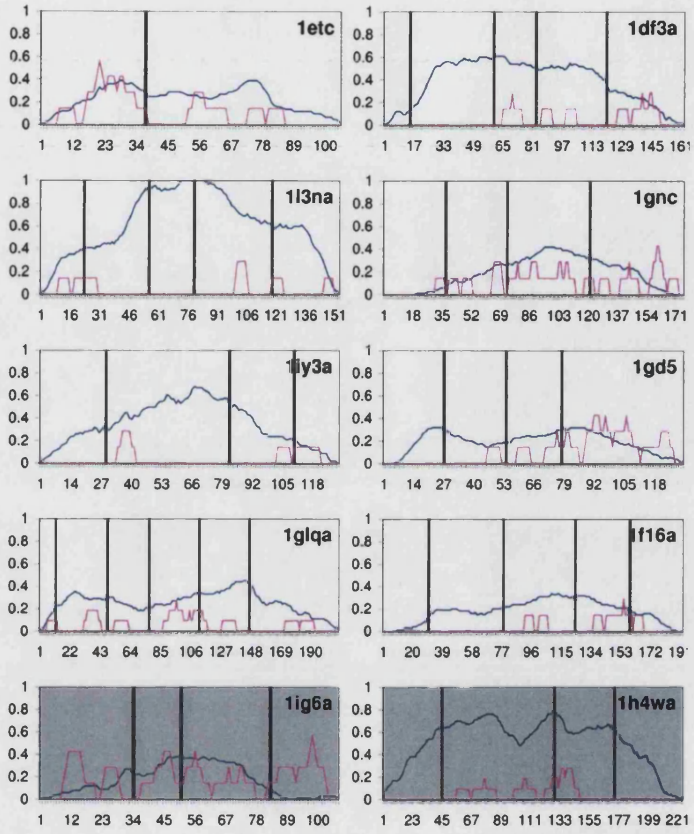


Figure 4.4: Frequency of crossover (pink) and tertiary contacts (blue) along the primary sequence of 10 human and mouse proteins (continuation of 4.3). The Y-axis shows the observed frequency of crossover in each of the evolving protein populations and the number of contacts divided by the length of the protein. The X-axis represents the amino acid sequence of each protein. Two examples explained in the text are shaded.

4.3 Implications for protein design

One of the main assumptions of this work is that introns are involved in the duplication, deletion and insertion of exons, and in the generation of chimeric protein-coding genes, as reviewed by Patthy (1999). Therefore, the fact that IEBs tend to exist away from artificial crossover hot spots could be applied to engineer proteins where one may want to insert fragments or to design chimeras. In silico recombination experiments could help in this task. In some cases, such as 1iy3a (see Figure 4.4), artificial crossover regions are highly

localised. Where this occurs, the information retrieved from these experiments is of little use, since large sections of the polypeptide have not been properly sampled. In other cases, such as 1bc9 or 1df3a (see Figures 4.3 and 4.4), recombination hot spots are well spread along the primary structure and their distribution could really help in the search for potential IEBs. This could be used to locate putative places for intron insertion within proteins that may have lost them, such as prokaryotic or even artificial proteins.

It is not clear if the difference in the distribution of artificial and natural crossover points is a property of proteins or just a consequence of the way the recombination algorithm works. Nevertheless the output of these simulations could be useful, especially when natural proteins show that introns can occur in any secondary structure environment and simple rules, despite the enrichment in coils observed in our data, have not been found.

Two examples in which artificial recombination was applied are now explained in more detail, with the aim of illustrating the relative importance of natural and artificially selected crossover points and to show how close IEB residues can be to functional sites.

4.3.1 Example 1: human Mrf-2 DNA-binding motif

Several structural studies on this protein (Yuan *et al.*, 1998; Whitson *et al.*, 1999; Zhu *et al.*, 2001) and its homologous sequences allowed us to build comparative models for all of them and perform artificial protein recombination, generating a profile as shown in Figure 4.4 (1ig6a). This protein specifically recognises a DNA sequence through helix 5 (major groove, see H5 in Figure 4.5) and two loops (minor groove, L1 and major groove, L2). Note that the frequency of crossover where natural introns are contained in the gene (numbered 1, 2, 3) is low. This result could help in the task of designing a composite transcription factor by showing which regions are more spatially constrained across evolution and which are less likely to disrupt the fold if modified. In this case, the N-terminal part of the L1 DNA-recognition loop is positively selected as a possible crossover point and it is this region that is predicted to interact with DNA (Zhu *et al.*, 2001). The C-terminal part of this loop appears not to interact with DNA but it is an integral part of the fold; thus changes here could impact directly on the fold stability and hence function. On the same lines, variability could be introduced into the major groove recognising helix (H5), where boundary 3 is located. However, recombining in these blue regions, e.g. near natural boundaries, could potentially cause a loss of function.

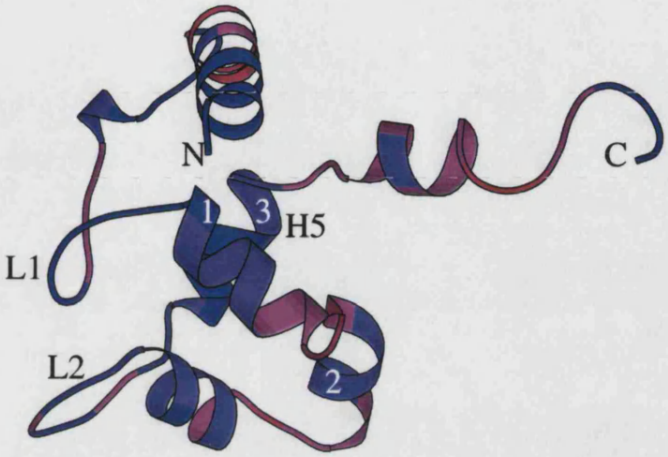


Figure 4.5: Protein recombination profile of human Mrf-2 DNA-binding domain mapped onto its three-dimensions model (1ig6a in Figure 4.4). N and C-termini are labelled. Helix 5 (H5) and loop 2 (L2) interact with the major groove of DNA, L1 with the minor groove. Introns found in the corresponding human gene are numbered 1, 2, and 3. Frequency of recombination is mapped to the protein backbone and represented as a colour gradient. Regions close to red are positively selected as crossover points, points that anchor recombination events and improve the fitness of the fold. Blue regions were not selected in the simulation. This diagram was prepared using Rasmol (Sayle & Milner-White, 1995) and Molscript (Kraulis, 1991).

4.3.2 Example 2: human brain trypsin

This example was chosen because it is an enzyme containing three IEBs, marked as 1, 2 and 3 (see Figure 4.6). Two of them are in close proximity (< 7Å) to the catalytic site, occupied in the figure by an inhibitor, as found in the PDB (Katona *et al.*, 2002). A total of 14 PDB templates were used to build comparative models, with sequence identities ranging from 38 to 100%, and these were subsequently recombined (see the profile in Figure 4.4, 1h4wa). The frequency of crossover along the sequence is shown by the variability of the colour of the backbone in Figure 4.6. Note that most of the recorded crossover events are at the surface of the protein, away from the binding pocket, in places that, nevertheless, affect the specificity of the enzyme (Perona & Craik, 1997). Unlike 1 and 3, boundary 2 is very close to an artificial recombination hot spot and stands more than 10Å away from the catalytic site. The four exons that build up this protein are shown

in Figure 4.7 with different colours. Clearly the binding site is the result of the precise packing of at least three exons and thus recombining at the boundaries between these exons (1 and 3) could be directly deleterious to the protein's function.

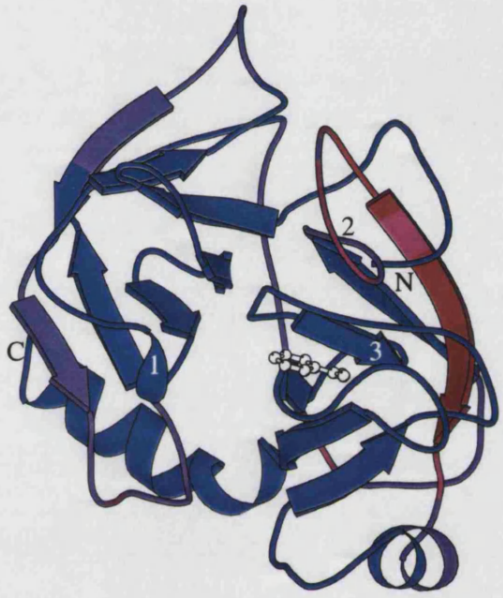


Figure 4.6: Protein recombination profile of human brain trypsin mapped onto its three-dimensional model (1h4wa in Figure 4.4), using the same colour scheme as in Figure 4.5. Intron boundaries are labelled 1, 2, and 3, as well as the N and C termini. An inhibitor to the active site, as deposited in the PDB(Katona *et al.*, 2002), is shown in white. This diagram was prepared using Rasmol (Sayle & Milner-White, 1995) and Molscript (Kraulis, 1991).

4.4 Discussion

In higher eukaryotes gene coding regions tend to be a small proportion of the genes, hence there is a higher probability of natural recombination events occurring at non-coding regions, including introns. In the context of the protein fold, introns could be acting as buffer regions that accommodate exon packing upon natural recombination, or even for accommodating entirely new domains. However, in our artificial recombination simulations we observe the opposite; crossover hot spots seem to steer away from intron boundaries. This is probably a consequence of the superimposition-based method used for our

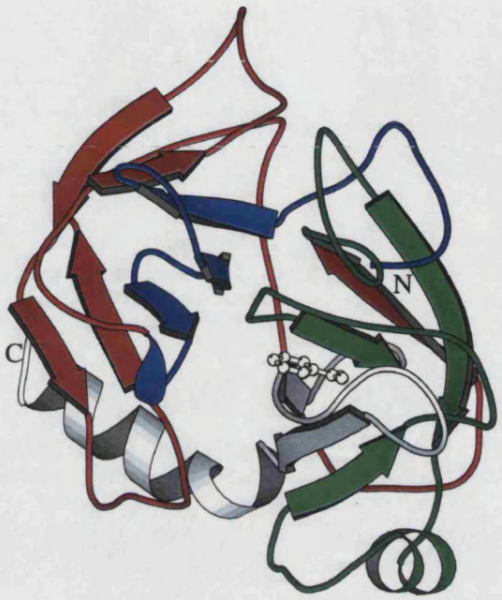


Figure 4.7: Exon structure of human brain trypsin, with 4 exons identified by different colours, showing that a close coordination between them is needed to form the active site.

recombination but also of the complex packing between exons in some cases (as shown in Figure 4.7). Because we treat protein fragments as rigid bodies we cannot simulate this accommodation. Perhaps by using protein docking techniques involving some flexibility we will be able to successfully recombine in virtually all parts of the protein, but at the cost of not longer being able to highlight natural IEBs. Thus the coarseness of our current approach may actually be an advantage.

The data presented here suggests an evolutionary feedback mechanism between natural introns and the effect they have on protein folds. Although there seems to be an enrichment of IEBs in coils and the ends of secondary structure elements, some natural introns occur at the midpoints of α-helices or β-strands. Therefore, in the task of designing protein recombination experiments, it is not possible to rule out regions according to their secondary structure. More complex criteria, such as protein structural fitness, tested here, may be needed. This is a stability criterion, which is not necessarily correlated to function. If function is to be modified or selected, extra restraints (or complementary functional experiments) should be required in the optimisation procedure. Indeed, recently Petersen & Taylor (2003) applied this sort of ideas to the design of novel zinc

binding proteins.

The statistical analysis performed here could also be useful to improve our protein recombination protocol. In particular, after this work, it seems necessary to allow the genetic algorithm to recombine proteins regardless of the secondary structure state of the residues involved, not just in loops. In addition, if flexibility is added, it could be useful to positively discriminate for intron boundary residues (when known) during recombination simulations. Nevertheless, currently it seems perhaps surprisingly difficult to successfully recombine on boundary regions, pointing to the possibility that crossing-over here may affect more dramatically protein folds, as measured with our fitness function.

What is the importance of IEB residues for protein specificity? From our data it seems that they are not more important in order to modify protein specificity than other residues in a protein; perhaps artificial crossover hot spots should be considered for these tasks. Furthermore, while usually there are only a few IEBs in a protein, artificial recombination protocols such as the one tested here may point out larger subsets of residues that are structurally conserved between homologous structures.

Finally, in relation to the introns early/late debate, our findings cannot exclude either theory. Some results seem to support an early origin of introns (such as secondary structure preferences) whilst others could be taken as evidence for their late origin (both packing and flexibility results). Our results seem to agree with a model in which both theories are compatible.

4.5 Conclusions

In this Chapter we did a series of statistical analyses and simulations approaching an evolutionary problem: the relation between the exonic scaffold of genes and the tertiary structure of the proteins that they code. In particular we look at the exonic components of folds, avoiding multi-domain proteins. In this analysis we learnt that introns do not populate randomly the genes in which they live, especially when protein secondary structure is considered. Their possible links to protein function were also explored, but our results suggest that the distribution of IEBs within protein folds is not affected significantly by their proximity to functional sites. Trying to investigate if these findings could be used in protein design, we generated protein crossover profiles and correlated them to protein function and structure. A weak negative correlation is found between natural IEBS and artificial crossover hot spots. In addition, it seems that crossover profiles can be useful to highlight regions related to enzymatic specificity or segments in protein folds where recombination events are more likely to be successful. These later findings will need

experimental studies in the laboratory to be fully appreciated. If there is a clear rule of thumb resulting from this work is that nature prefers to put IEBs in loops, and so protein engineers should use loops to make substantial modifications of protein folds or to make chimeras.

4.6 Problems and possible developments

The analysis done in this chapter shows some interesting data, but nevertheless some problems were found. To further the extent of the analysis the following points should be considered:

- The protocol for protein recombination, used here and explained in the previous chapter, is a way of avoiding the step of selecting templates for Comparative Modelling. However, the goal here is not building models, it is to highlight certain regions along proteins. Here templates from different origins are mixed in a pool of models with the aim of obtaining artificial recombination profiles. How much the initial composition of the pool determines the outcome of the recombination simulations is a question that we have not answered. Further work in this direction may be important, since this issue could directly affect the sampling of crossover points along a protein fold.

- As explained in Chapter 3, the recombination protocol is non-deterministic and therefore it can generate different outputs for the same input. For this reason, given more computing time, it will be necessary to compare different recombination runs for the same initial pool of CM models and analyse the differences and the similarities between their recombination profiles. This is important since the negative correlation found in this work between the placement of IEBs and crossover hot spots is weak. Perhaps building consensus crossover profiles would be a good idea. However, this correlation could also be genuinely weak or even just a consequence of the dataset used here. In either case, these seem to be important things to check in future work. Nevertheless, the data presented here suggests that, even in the absence of a clear correlation, crossover profiles should be useful for protein design studies, such as changing the specificity of enzymes.

- An important limitation of the recombination procedure tested here is the need for different templates to generate an initial pool of models. At this moment in time we do not know if alternative structures for the same molecule, such as those extracted

from different experimental conditions or techniques (crystallography,NMR or even MD), could also be used.

- In Section 4.4 we discussed the possibility of introducing flexibility into the crossover mechanism, by perhaps using protein docking techniques. This remains to be done.
- The data shown here (see Section 4.1.1) suggests certain preferences at the secondary structure level for the placement of IEBs. This preferences could be used to bias the occurrence of crossover events along a protein's sequence.

4.7 Materials and Methods

Datasets. The protein set used throughout this work was composed of human and mouse proteins obtained from the Protein Data Bank (PDB, as of 22nd January 2003). To avoid large multidomain proteins, only structures with at least 100 residues but no more than 300 were selected. To avoid spliced genes, immunoglobulins and T-cell receptors were identified by sequence similarity and excluded from this dataset. Chimeric proteins were also excluded. After excluding proteins with only one exon (about 25% of the original set), this dataset contained a total of 684 PDB chains. These proteins contain, on average, 3.2 introns. For the study of human-mouse homologous proteins, human and mouse sequence pairs of sequence identity above 40% were extracted from the above dataset, resulting in 118 pairs. Many homologous sequences are contained in this set but no effort was made to remove redundancy, since it was observed that almost identical proteins may have a different number of introns, in different positions along the sequence.

A subset of 22 proteins (shown in Table 4.5), selected to cover different folds and functions was used to perform recombination experiments with comparative models built from both evolutionary close and remote homologous structures in the PDB. These 22 proteins were selected to avoid multi-domain proteins, and have diverse comparative modelling templates that could be confidently aligned.

Assignment of introns. Intron boundaries were assigned by mapping protein sequences to the human (NCBI Human Contig Assembly 31, November 2002 freeze) and murine (MGSCv3 release 3, February 2002 freeze34) genome assemblies, using the BLAT server (Kent, 2002). When using protein amino acid sequences in this work, IEBs are defined as the residues corresponding to the left side boundary at the DNA level. Introns in homologous proteins are said to be conserved if they occupy exactly the same place in the structural alignment of those proteins.

Secondary structure, comparative modelling. Protein secondary-structure was assigned using the program DSSP (Kabsch & Sander, 1983). Comparative protein models were built using our server 3D-JIGSAW in the interactive mode, using alignments with bit-scores of at least 1.8 and as many different templates as possible. Some templates were extracted from the corresponding PFAM families (see Table 4.5) using DomainFishing.

Calculation of contacts. To calculate the tertiary contacts around a given residue r, every C_β from residues to the left of r was checked against every C_β to the right in the protein sequence, calculated in a similar fashion to Voigt *et al.* (2002). A contact was then defined as a pair of C_β separated less than 7.0Å in Cartesian space and more than 4 residues in sequence, as previously described (Hu *et al.*, 2002).

Recombination of proteins. The protein recombination protocol used is a modification of the one previously described (see Section 3.10), that adds new side chains in every mutation event using the program SCWRL (Dunbrack & Karplus, 1993) and performs up to 5 rounds of steepest descent minimisation on every newly created sibling (the C code for this minimisation protocol was written by Paul W.Fitzjohn). Crossover events are not restricted to any secondary structure state, since it was observed that, despite some preferences, natural IEB boundaries can be located in any context. Sampled artificial crossover points were recorded in real time (in the PDB format B-factor column) to be later analysed and create the profiles shown in Figures 4.3 and 4.4.

Local flexibility at intron-exon boundaries. A subset (118) of homologous human-mouse pairs with pairwise sequence identity over 40% was extracted from our original dataset. IEBs were mapped onto PDB structures and each of the human-mouse sequence pairs superimposed using *msuper*. A window of seven residues was moved along the superposition and the fitness of the alignment recorded by summing *msuper* alignment scores, ranging from 0 for a good fit to 9 for a bad fit (C_β-C_β distances, in Å), for each of the seven positions. The DSSP program was used to assign the secondary structure elements for aligned sequences and residues classified as participating in a strand, helix or coil region. The window scores for each of the three secondary structure elements were then normalised and the scores for IEBs were compared to the overall expected scores.

Packing of exons using structural alignments. The average exon length in the dataset of 118 human and mouse sequence pairs of sequence identity over 40% was calculated

(41 residues). This value was increased by 5% to compensate for alignment gaps between the pairs, bringing the exon length to 43. Pairs of PDB sequences were aligned using *Clustalw* and pairs containing more than 20% alignment gaps were excluded. Two adjacent windows of the average exon length, representing two theoretical exons, were moved along the aligned sequence pair and a structural alignment performed using *msuper*, superimposing the two left hand exons on each other and carrying over the structure of the right hand exons as rigid bodies (see Figure 4.1). A vector from the N-terminus of the right hand exon to the centre of geometry of the same exon was calculated for both sequences and the angle between the vectors determined. This was repeated for the whole length of the sequence alignment. Sequence alignments too short to yield at least 30 angles were excluded, lowering the total number of pairs to 112. These distributions of angles were then normalised so that they could be added to create the overall normalised distributions shown in Figure 4.1.

Chapter 5

Concluding remarks

Three years of work have been condensed in the previous pages. Here I will try to summarise the results obtained and to put them together in a biological context. What does this work contribute to our biological knowledge? How does this work add to the repertoire of computational tools used in molecular Biology?

In **Chapter 2** we found that, as far as protein Comparative Modelling is concerned, none of our sequence alignment techniques can be considered to be perfect and although it is possible to rank them, in certain situations 'weaker' techniques can perform better than 'stronger' ones. Despite these limitations, we designed tools for defining protein domains, finding structural templates and aligning them. We also explored evaluators of alignment quality, such as *bit-scores* or 3D-conservation maps. Following our analysis of alignment methods, in **Chapter 3** we found that, in agreement with observations in the literature, it is not trivial to select alignments and templates for building a comparative model. Motivated by this we explored a new way of combining this data, by using a genetic algorithm based on natural genetic recombination. In addition to the genetic algorithm itself, this *in silico* recombination protocol borrowed many of its algorithmic components, such as dynamic programming, secondary structure assignment or the estimation of protein stability. This recycled set of tools, arranged in this particular way, seems to be able to construct protein models in a robust manner, with the ability to resolve at least some alignment conflicts and therefore correct errors. The program is able to produce alternative but similar protein structures for the same amino acid sequence, NMR-like ensembles. As our encouraging results (and others (Fischer, 2003)) in CASP5 suggest, this combinatorial approach can be equally useful for Fold Recognition purposes. Finally, in **Chapter 4** we applied this newly designed protein recombination methodology to approach an evolutionary problem: the relation between the exonic scaffold of genes

and the tertiary structure of the proteins that they code. In this analysis we learnt that introns do not populate randomly the genes in which they live, especially when protein secondary structure is considered. Their possible links to protein fitness and function were also explored. Trying to investigate if these findings could be used in protein design, we generated protein crossover profiles and correlated them to protein function and structure. While only a weak negative correlation is found between natural intron-exon boundaries and artificial crossover hot spots, crossover profiles can be useful to highlight regions related to enzymatic specificity or segments in protein folds where recombination events are more likely to be successful.

A set of tools has been developed during the course of this work, in the form of web servers, to assist the experimentalist. These tools are:

- DomainFishing (http://www.bmm.icnet.uk/~3djigsaw/dom_fish), linked to the comparative modelling server 3D-JIGSAW, where the user can define domains, find templates, align them and build protein models easily and interactively. Both servers are extensively used by the community and their performance can be monitored through the EVA automatic continuous evaluation. The overall performance of our approach (see Table 2.8), is promising as we are able to model difficult models without compromising the quality.

- *in silico* protein recombination, (http://www.bmm.icnet.uk/~3djigsaw/recomb), where the user can recombine a set of models obtained from different sources.

Appendix A

The program *msuper*

msuper stands for multiple structure superimposition and is a computer program written in C++ based on the published work of Russell & Barton (1992) and Gerstein & Levitt (1996). This is a progressive multiple structure alignment protocol, related to *Clustalw* in the sense that it only performs two-dimension dynamic programming and keeps updating the growing multiple structure profile as new structures are added. A Linux binary and some documentation can be found at: http://www.bmm.icnet.uk/~contrera/msuper/.

The cornerstone of the algorithm is the pairwise structural alignment routine, *Super* (see Section A.1). This routine includes a global dynamic programming subroutine (*struct_align*) in which the matrix is filled with the distances between every possible pair of C_β atoms of a couple of proteins p_1 and p_2. C_β atoms are preferred to C_α to minimise the chance of misalignments by one residue, especially in strands (Gerstein & Levitt, 1996). Instead of using evolutionary or probabilistic criteria to score matches, a simple Euclidean distance is taken. The squared distance values are scaled in the range [0-20] following the criterion used by Gerstein & Levitt (1996). For this range of values gap costs of 2.0 and 0.5 (opening and extension) are adequate.

The two proteins to be structurally aligned are first put in the same frame of reference, by correcting each atom's positions with respect to the protein's centroid. In addition, it is required that p_1 and p_2 are at least approximately superimposed so that equivalent residues in the pair of proteins come close in space. For *msuper* we used a linear least-squares minimisation routine written by Andras Aszodi implementing an algorithm published by McLachlan (1979). This routine uses the Singular Value Decomposition (SVD) algebraic method (see for example Gershenfeld (1999)), that minimises the RMSD (Equation 3.5) between two equally sized sets of points. In our algorithm, these sets of points are the equivalent residues in a global alignment. It is this need for a seed alignment that

limits the applicability of *msuper* for cases with very low sequence similarity. If the seed alignment is significantly wrong, the rest of the algorithm might not be able to produce a sensible structural alignment.

By iteratively aligning in distance space and superimposing, the RMSD between p_1 and p_2 usually converges and the final structural alignment is obtained. Following Russell & Barton (1992), the raw dynamic programming score *DPscore* of the alignment A is corrected by considering the amount of insertions and deletions introduced:

$$Score(A) = \frac{DPscore(A)}{Length(A)} \cdot \frac{Length(A) - gaps(p_1)}{Length(p_1)} \cdot \frac{Length(A) - gaps(p_2)}{Length(p_2)} \qquad (A.1)$$

A.1 Algorithm details

The most important part of the program is the *Super* routine, which is now outlined:

```
Super( Alignment &Ali , Protein &p1 , Protein &p2 )
{
    /* Ali is the seed sequence alignment */
    rmsd = SVD( p1 , p2 , rotationMatrix , Ali ); /* Singular Value Decomp, see text */
    Alignment thriD = p1→struct_align( p2 );
    while(|rmsd - previous_rmsd| > 0.005) && (rounds<MaxRounds))
    {
        rmsd = SVD( p1 , p2 , rotationMatrix , thriD );
        thriD = p1→struct_align( p2 );
        rounds++;
    }
    /* apply final rotation matrix to superimposed p1 */
    p1→apply_rotation_matrix( rotationMatrix );
    return thriD; /* return final sequence alignment */
}
```

This is the *msuper* algorithm:

```
main( FILE input_file )
{
  while (input_file) /* read input file and the corresponding PDB files */
  {
    prot = new Protein(PDB_file); /* create Protein object from read file */
    prot→readPDB_and_DSSP();
    prot→checkPDB(); /* check PDB & add Cbeta to Glycines */
    prot_list.push_back( prot );
  }

  /* all vs. all pairwise alignments */
  for(i=0;i<prot_list.size();i++)
  {
    for(j=i+1;j<prot_list.size();j++)
    {
      Alignment Seq_pair = prot_list[i]→Sequence+SS_align( prot_list[j] , BLOSUM );
      Alignment Str_pair = prot_list[i]→Super( Seq_pair, prot_list[j] );
      ali_stock.push_back( Str_pair );
    }
  }
  /* rank proteins by their accumulated pairwise scores */
  sort_list_proteins( &ali_stock , &prot_list );

  /* Start progressive global multiple structural alignment */
  mult = new MultipleAlignment( prot_list[0] );
  /* mult computes an average Cbeta pseudoprotein as new structures are added */
  for(i=1;i<prot_list.size();i++)
  {
    Alignment sup = mult.pseudoprotein→Super( ali_stock[i], prot_list[i] );
    if(sup.sscore() < bad_struct_score ) break; /* stop growing multiple alignment */
    mult→add_ali( sup, prot_list[i] );
  }
}
```

A.2 Comparison to SSAP and example

As in Section 2.2.1, a set of 317 pairs of homologous SCOP domains was used to compare *Clustalw* and *Profile1* pairwise alignments to both SSAP and *msuper* structural alignments. To evaluate alignments the same shift-score was used (see Section 2.2). When comparing the shift scores obtained with respect to *msuper* alignments to those obtained with SSAP, linear correlation coefficients of 0.86 (*Clustalw*) and 0.84 (*Profile1*) were obtained. A graphical representation of these results is shown in Figure A.1.

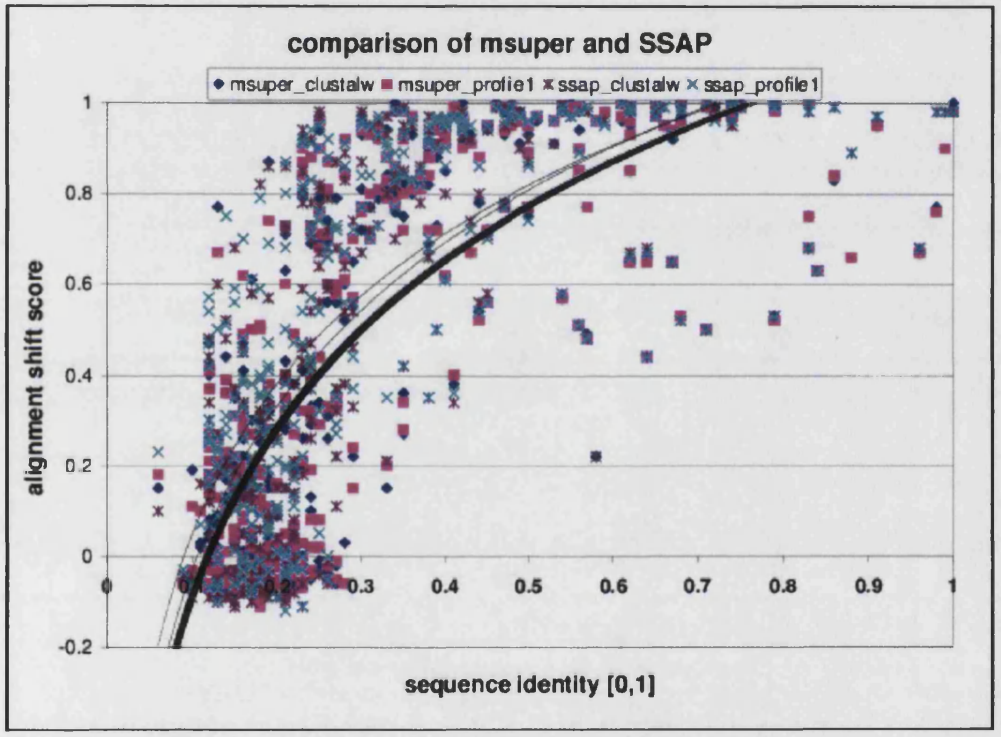


Figure A.1: Comparison of *msuper* and SSAP reference alignments with respect to *Clustalw* and *Profile1* pairwise alignments. Logarithmic fits are also shown, with the thin lines corresponding to the SSAP series and the thick to *msuper*. Note that *msuper* appears to give a closer match to sequence alignments, for the same shift error, than SSAP. This is probably a consequence of the fact that the initial seed alignment for *msuper* is sequence-based.

An example of an alignment comparison is given in Table A.1.

	SSAP:
1d5ya	EFTMPEHKFVTLEDTPLIGVTQSYSCSLEQISDFRHEMRYQFWHDFLGNAPTIPPVLYGL
1bowa	--RLGEVFVLDEEEIRIIQTEAEG---------IGPENVLNASYSKLKKFIESNNSYGAT
1d5ya	NETRPSQDKDDEQEVFYTTALAQDQADGYVLTGHPVMLQGGEYVMFTYEGLGTGVQEFIL
1bowa	FSFQPYTSIDE--MTYRHIFTPVL-ISSITPDMEITTIPKGRYACIAYNFSPEHYFLNLQ
1d5ya	TVYGTCMPMLNLTRRKGQDIERYYPAEDDRPINLRCELLIPIR
1bowa	KLI-KYIADRQLTVV-SDVYELIIPIH----YEYRVEMKIRIL
	***msuper*:**
1d5ya	EFTMPEHKFVTLEDTPLIGVTQSYSCSLEQ-ISDFRHEMRYQFWHDFLGNAPTIPPVLYG
1bowa	--RLGEVFVLDEEEIRIIQTEAEGIG--PENVLNASYSKLKKFI-ES-------NNSYGA
1d5ya	LNETRPSQDKDDEQEVFY-TTALAQDQADGYVLTGHPVMLQGGEYVMFTYEGLGTGVQEF
1bowa	TF-SFQP-YTSIDEMT-YRHIFTPVL-ISSITPDMEITTIPKGRYACIAYN--F-S-PEH
1d5ya	ILTVYGTC-MPML-NL-TRRKGQDIERYYPAEDDRPINLRCELLIPIRRKLAAA
1bowa	YFLNLQ-KLIKYIADRQLTVVSDVYELIIP-IH---YEYRVEMKIRIL------
	shift score calculation between the two methods:
1d5ya	EFTMPEHKFVTLEDTPLIGVTQSYSCSLEQ-ISDFRHEMRYQFWHDFLGNAPTIPPVLY
1bowa	--RLGEVFVLDEEEIRIIQTEAEGIG--PENVLNASYSKLKKFI-ES-------NNSYG
SS	CCCCCCEEEEEECCEEEEEEEEECCCCHHH-HHHHHHHHHHHHHHHHHHCCCCCCCEE
SS	--CCCCEEEEEECCEEEEEECCCCC--HHHHCCCCCHHHHHHC-CC-------CCCEE
shift	..00000000.....000000000..........8888777777.77.......00000
1d5ya	GLNETRPSQDKDDEQEVFY-TTALAQDQADGYVLTGHPVMLQGGEYVMFTYEGLGTGVQ
1bowa	ATF-SFQP-YTSIDEMT-YRHIFTPVL-ISSITPDMEITTIPKGRYACIAYN--F-S-P
SS	EEEEEEECCCCCCCEEEEE-EEEEEHHHHHHHCCCCEEEEECCEEEEEEEEEEEHHHHH
SS	EEE-CCCC-CCCCCCCC-CCEEEEECC-CCCCCCCCEEEEECCEEEEEEEE--C-C-H
shift	000.0011.1111..00.0.0000000.00000000000000000000000..1.2.3
1d5ya	EFILTVYGTC-MPML-NL-TRRKGQDIERYYPAEDDRPINLRCELLIPIRRKLAAA
1bowa	EHYFLNLQ-KLIKYIADRQLTVVSDVYELIIP-IH---YEYRVEMKIRIL------
SS	HHHHHHHHCH-HHHC-CC-EECCCCEEEEECHHHCCCCCEEEEEEEEEEECCCCCC
SS	HHHHHHHH-HHHHHHHHHHCCEEEEEEEEEEE-CC---CCEEEEEEEEC------
shift	3333333..3.3222.21.110.000000000.0....0000000000........

Table A.1: Alignment comparison of the pair 1d5ya.1bowa, yielding a total shift score of 0.58 between the two methods. The shift score was calculated as derived by Cline (2000). Note that the sequence identity between these two proteins is below 15%.

Appendix B

Internet resources used

Table B.1: URLs for some Internet resources mentioned or used within this work.

URL	Description
ftp://ftp.ncbi.nih.gov/blast	BLAST, PSI-BLAST and IMPALA executable programs
ftp://ftp.ncbi.nih.gov/blast/db/nr.Z	non-redundant protein sequence database
ftp://ftp.ncbi.nih.gov/pub/seg/	software to detect low complexity regions in protein sequences
http://www.sbg.bio.ic.ac.uk/3dgenomics	Comparison of Genomes via Protein Structure
http://astral.stanford.edu/	protein sequences for SCOP domains
http://bioinf.cs.ucl.ac.uk/psipred/	secondary structure prediction of protein sequences
http://salilab.org/~eva	EValuation of Automatic protein structure prediction
http://hmmer.wustl.edu/	HMMer software package for hidden Markov models
http://scop.mrc-lmb.cam.ac.uk/scop/	Structural Classification Of Proteins
http://www.biochem.ucl.ac.uk/bsm/cath/	CATH structural classification of proteins
http://www.bmm.icnet.uk	Biomolecular Modelling site at Cancer Research UK
http://www.ncbi.nlm.nih.gov	U.S. National Center for Biotechnology Information
http://www.ncbi.nlm.nih.gov/BLAST/	interactive BLAST and PSI-BLAST
http://www.openpbs.org	load sharing system for distributed processing

(continued on next page)

Table B.1: URLs for some Internet resources mentioned or used within this work.

(continued from previous page)

URL	Description
http://www.rcsb.org/	Protein Data Bank
http://www.sanger.ac.uk/Software/Pfam	PFAM, protein family and domain database
http://www.sbg.bio.ic.ac.uk/3dpssm/	remote homology detection of protein of known structure
http://www.structuralgenomics.org/	resource for structural genomics
http://expasy.org/sprot/	Swiss-Prot Protein knowledgebase
http://www.ensembl.org	Ensembl Genome Browser
http://wolf.bms.umist.ac.uk/naccess	NACCESS
http://predictioncenter.llnl.gov/casp5	CASP5
http://www.bmm.icnet.uk/~3djigsaw	3D-JIGSAW
http://www.sbg.bio.ic.ac.uk/~3dpssm	3D-PSSM
http://alax.bio.nagoya-u.ac.jp	Alax
http://www.gmd.de/SCAI	Arby
http://www.bmm.icnet.uk/~3djigsaw/dom_fish	DomainFishing
http://www.fundp.ac.be/urbm/bioinfo/esypred	EsyPred3D
http://physchem.pharm.kitasatou.ac.jp	FAMS
http://www-cryst.bioc.cam.ac.uk/~fugue	FUGUE
http://www.cs.bgu.ac.il/~bioinbgu	INBGU
http://PredictionCenter.llnl.gov/local/lga	LGA
http://www.sbc.su.se/~arne/pcons	Pmodeller
http://www.cs.bgu.ac.il/~dfischer/CAFASP3	CAFASP
http://www.sbg.bio.ic.ac.uk/~mueller/TeXMed/	TeXMed - a BibTeX interface for PubMed
http://www.google.com	Google search tools

Appendix C

Papers published during this project

Most of the work described here has been published as part of articles in peer-reviewed journals. These articles are sorted here in chronological order:

- Contreras-Moreira, B. & P. A. Bates (2002). Domain Fishing: a first step in protein comparative modelling. *Bioinformatics*, 18(8):1141–2.
- Contreras-Moreira, B., P. W. Fitzjohn & P. A. Bates (2002). Comparative modelling: an essential methodology for protein structure prediction in the post-genomic era. *Applied Bioinformatics*, 1(4):177–190.
- Contreras-Moreira, B., P. W. Fitzjohn & P. A. Bates (2003). In silico Protein Recombination: enhancing template and sequence alignment selection for comparative protein modelling. *Journal of Molecular Biology*, 328:593–608.
- Contreras-Moreira, B., P. A. Jonsson & P. A. Bates (2003). Structural context of exons in protein domains: implications for protein modelling and design. *Journal of Molecular Biology*, 333:1057-1071.
- Contreras-Moreira, B., P. W. Fitzjohn, M. Offman, G. R. Smith & P. A. Bates (2003). Novel use of a genetic algorithm for protein structure prediction: searching template and sequence alignment space. *Proteins*, S6:424-429.

References

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. & Watson, J. D. (**1994**). *Molecular Biology of the Cell.* Garland Pub., New York, 3rd edition.

Allen, M., Friedler, A., Schon, O. & Bycroft, M. (**2002**). The structure of an FF domain from human HYPA/FBP11. *J.Mol.Biol.*, 323(3):411–416.

Aloy, P., Ciccarelli, F. D., Leutwein, C., Gavin, A. C., Superti-Furga, G., Bork, P., Bottcher, B. & Russell, R. B. (**2002**). A complex prediction: three-dimensional model of the yeast exosome. *EMBO Rep.*, 3(7):628–635.

Altschul, S. F. & Gish, W. (**1996**). Local alignment statistics. *Methods Enzymol.*, 266:460–480.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (**1990**). Basic local alignment search tool. *J.Mol.Biol.*, 215(3):403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (**1997**). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17):3389–3402.

Anfinsen, C. B., Haber, E., Sela, M. & White Jr, F. (**1961**). The kynetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *PNAS*, 47(9):1309–1314.

Aramini, J. M., Mills, J. L., Xiao, R., Acton, T. B., Wu, M. J., Szyperski, T. & Montelione, G. T. (**2003**). Resonance assignments for the hypothetical protein yggU from Escherichia coli. *J.Biomol.NMR*, 27(3):285–286. Letter.

Attwood, T. K., Beck, M. E., Flower, D. R., Scordis, P. & Selley, J. N. (**1998**). The PRINTS protein fingerprint database in its fifth year. *Nucleic Acids Res.*, 26(1):304–308.

Baker, D. & Sali, A. (**2001**). Protein structure prediction and structural genomics. *Science*, 294(5540):93–96.

Baldi, P., Chauvin, Y., Hunkapiller, T. & McClure, M. A. (**1994**). Hidden Markov models of biological primary sequence information. *PNAS*, 91(3):1059–1063.

Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S. R., Griffiths-Jones, S., Howe, K. L., Marshall, M. & Sonnhammer, E. L. (**2002**). The Pfam protein families database. *Nucleic Acids Res.*, 30(1):276–280.

Bates, P. A., Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (**2001**). Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins*, Suppl(5):39–46.

Bates, P. A. & Sternberg, M. J. (**1999**). Model building by comparison at CASP3: Using expert knowledge and computer automation. *Proteins*, 37(S3):47–54.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., Rapp, B. A. & Wheeler, D. L. (**2002**). GenBank. *Nucl. Acids Res.*, 30(1):17–20. URL `http://nar.oupjournals.org/cgi/content/abstract/30/1/17`.

Berezovsky, I. N., Grosberg, A. Y. & Trifonov, E. N. (**2000**). Closed loops of nearly standard size: common basic element of protein structure. *FEBS Lett.*, 466(2-3):283–286.

Berezovsky, I. N. & Trifonov, E. N. (**2001**). Van der Waals locks: loop-n-lock structure of globular proteins. *J.Mol.Biol.*, 307(5):1419–1426.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (**2000**). The Protein Data Bank. *Nucleic Acids Res.*, 28(1):235–242.

Betts, M. J., Guigo, R., Agarwal, P. & Russell, R. B. (**2001**). Exon structure conservation despite low sequence similarity: a relic of dramatic events in evolution. *EMBO J.*, 20(19):5354–5360.

Blundell, T. L., Sibanda, B. L., Sternberg, M. J. & Thornton, J. M. (**1987**). Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, 326(6111):347–352.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S. & Schneider, M. (**2003**). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, 31(1):365–370.

Bond, C. J., Huang, J., Hajduk, R., Flick, K. E., Heath, P. J. & Stoddard, B. L. (**2000**). Cloning, sequence and crystallographic structure of recombinant iron superoxide dismutase from Pseudomonas ovalis. *Acta Crystallogr. D Biol. Crystallogr.*, 56 (Pt 11):1359–1366.

Bower, M. J., Cohen, F. E. & Dunbrack, R. L., J. (**1997**). Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *J.Mol.Biol.*, 267(5):1268–1282.

Bowie, J. U., Luthy, R. & Eisenberg, D. (**1991**). A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253(5016):164–170.

Branden, C.-I. & Tooze, J. (**1999**). *Introduction to protein structure*. Garland Pub., New York, 2nd edition.

Braun, W. & Go, N. (**1985**). Calculation of protein conformations by proton-proton distance constraints. A new efficient algorithm. *J.Mol.Biol.*, 186(3):611–626.

Brenner, S. E., Koehl, P. & Levitt, M. (**2000**). The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res.*, 28(1):254–256.

Brodersen, D. E., Clemons, W. M., J., Carter, A. P., Wimberly, B. T. & Ramakrishnan, V. (**2002**). Crystal structure of the 30 S ribosomal subunit from Thermus thermophilus: structure of the proteins and their interactions with 16 S RNA. *J.Mol.Biol.*, 316(3):725–768.

Broo, K., Larsson, A. K., Jemth, P. & Mannervik, B. (**2002**). An ensemble of theta class glutathione transferases with novel catalytic properties generated by stochastic recombination of fragments of two mammalian enzymes. *J.Mol.Biol.*, 318(1):59–70.

Brooks, B., Bruccoleri, R., Olafson, B., States, D., Swaminathan, S. & Karplus, M. (**1983**). CHARMM: A program for macromolecular energy, minimization and dynamics calculation. *J. Comp. Chem.*, 4:187–217.

Bujnicki, J. M., Elofsson, A., Fischer, D. & Rychlewski, L. (**2001**). LiveBench-2: large-scale automated evaluation of protein structure prediction servers. *Proteins*, Suppl(5):184–191.

Burley, S. K. (**2000**). An overview of structural genomics. *Nat.Struct.Biol.*, 7 Suppl:932–934.

Cedergren-Zeppezauer, E. S., Goonesekere, N. C., Rozycki, M. D., Myslik, J. C., Dauter, Z., Lindberg, U. & Schutt, C. E. (**1994**). Crystallization and structure determination of bovine profilin at 2.0 A resolution. *J.Mol.Biol.*, 240(5):459–475.

Chen, Y. W., Bycroft, M. & Wong, K.-B. (**2003**). Crystal structure of ribosomal protein L30e from the extreme thermophile Thermococcus celer: thermal stability and RNA binding. *Biochemistry*, 42(10):2857–2865.

Chothia, C., Gough, J., Vogel, C. & Teichmann, S. A. (**2003**). Evolution of the protein repertoire. *Science*, 300(5626):1701–1703.

Chothia, C. & Lesk, A. M. (**1986**). The relation between the divergence of sequence and structure in proteins. *EMBO J.*, 5(4):823–826.

Clamp, M., Andrews, D., Barker, D., Bevan, P., Cameron, G., Chen, Y., Clark, L., Cox, T., Cuff, J., Curwen, V., Down, T., Durbin, R., Eyras, E., Gilbert, J., Hammond, M., Hubbard, T., Kasprzyk, A., Keefe, D., Lehvaslaiho, H., Iyer, V., Melsopp, C., Mongin, E., Pettett, R., Potter, S., Rust, A., Schmidt, E., Searle, S., Slater, G., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Stupka, E., Ureta-Vidal, A., Vastrik, I. & Birney, E. (**2003**). Ensembl 2002: accommodating comparative genomics. *Nucl. Acids Res.*, 31(1):38–42. URL `http://nar.oupjournals.org/cgi/content/abstract/31/1/38`.

Clark, F. & Thanaraj, T. A. (**2002**). Categorization and characterization of transcript-confirmed constitutively and alternatively spliced introns and exons from human. *Hum.Mol.Genet.*, 11(4):451–464.

Cline, M. (**2000**). *Protein sequence alignment reliability: prediction and measurement.*. Ph.D. Dissertation., University of California, Santa Cruz.

Colloc'h, N. & Cohen, F. E. (**1991**). Beta-breakers: an aperiodic secondary structure. *J.Mol.Biol.*, 221(2):603–613.

Contreras-Moreira, B. & Bates, P. A. (**2002**). Domain Fishing: a first step in protein comparative modelling. *Bioinformatics*, 18(8):1141–1142.

Contreras-Moreira, B., Fitzjohn, P. W. & Bates, P. A. (**2002**). Comparative modelling: an essential methodology for protein structure prediction in the post-genomic era. *Applied Bioinformatics*, 1(4):177–190.

Contreras-Moreira, B., Fitzjohn, P. W. & Bates, P. A. (**2003a**). In silico Protein Recombination: enhancing template and sequence alignment selection for comparative protein modelling. *J.Mol.Biol.*, 328:593–608.

Contreras-Moreira, B., Fitzjohn, P. W., Offman, M., Smith, G. R. & Bates, P. A. (**2003b**). Novel use of a genetic algorithm for protein structure prediction: searching template and sequence alignment space. *Proteins*, S6:424–429.

Contreras-Moreira, B., Jonsson, P. F. & Bates, P. A. (**2003c**). Structural context of exons in proteins domains: implications for protein modelling and design. *J.Mol.Biol.*, 333:1057–1071.

Cornell, W., Cieplak, P., Bayly, C., Gould, I., Merz, K. J., Ferguson, D., Spellmeyer, D., Fox, T., Caldwell, J. & Kollman, P. (**1995**). A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J.Am.Chem.Soc.*, 117:5179–5197.

Corpet, F., Servant, F., Gouzy, J. & Kahn, D. (**2000**). ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. *Nucleic Acids Res.*, 28(1):267–269.

Craik, C. S., Rutter, W. J. & Fletterick, R. (**1983**). Splice junctions: association with variation in protein structure. *Science*, 220(4602):1125–1129.

Craik, C. S., Sprang, S., Fletterick, R. & Rutter, W. J. (**1982**). Intron-exon splice junctions map at protein surfaces. *Nature*, 299(5879):180–182.

Damm, W., Frontera, A., Tirado-Rives, J. & Jorgensen, W. (**1997**). OPLS All-Atom Force Field for Carbohydrates. *J.Comp.Chem.*, 18:1955–1970.

de Groot, B. L., van Aalten, D. M., Scheek, R. M., Amadei, A., Vriend, G. & Berendsen, H. J. (**1997**). Prediction of protein conformational freedom from distance constraints. *Proteins*, 29(2):240–251.

De Maeyer, M., Desmet, J. & Lasters, I. (**2000**). The dead-end elimination theorem: mathematical aspects, implementation, optimizations, evaluation, and performance. *Methods Mol.Biol.*, 143:265–304.

de Souza., S. J., Long, M., Klein, R. J., Roy, S., Lin, S. & Gilbert, W. (**1998**). Toward a resolution of the introns early/late debate: only phase zero introns are correlated with the structure of ancient proteins. *PNAS*, 95(9):5094–5099.

de Souza., S. J., Long, M., Schoenbach, L., Roy, S. W. & Gilbert, W. (**1996**). Intron positions correlate with module boundaries in ancient proteins. *PNAS*, 93(25):14632–14636.

de Souza., S. J., Long, M., Schoenbach, L., Roy, S. W. & Gilbert, W. (**1997**). The correlation between introns and the three-dimensional structure of proteins. *Gene*, 205(1-2):141–144.

Desjarlais, J. R. & Handel, T. M. (**1999**). Side-chain and backbone flexibility in protein core design. *J.Mol.Biol.*, 290(1):305–318.

Desmet, J., De Maeyer, M., Hazes, B. & Lasters, I. (**1992**). The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542.

Devos, D., Garmendia, J., de Lorenzo, V. & Valencia, A. (**2002**). Deciphering the action of aromatic effectors on the prokaryotic enhancer-binding protein XylR: a structural model of its N-terminal domain. *Environ.Microbiol.*, 4(1):29–41.

Dill, K. A. (**1990**). The meaning of hydrophobicity. *Science*, 250(4978):297–298.

Dobson, C. M. & Karplus, M. (**1999**). The fundamentals of protein folding: bringing together theory and experiment. *Curr.Opin.Struct.Biol.*, 9(1):92–101.

Dunbrack, R. L., J. & Karplus, M. (**1993**). Backbone-dependent rotamer library for proteins. Application to side- chain prediction. *J.Mol.Biol.*, 230(2):543–574.

Eddy, S. R. (**1996**). Hidden Markov models. *Curr.Opin.Struct.Biol.*, 6(3):361–365.

Edwards, M. S., Sternberg, J. E. & Thornton, J. M. (**1987**). Structural and sequence patterns in the loops of beta alpha beta units. *Protein Eng.*, 1(3):173–181.

Efimov, A. V. (**1991**). Structure of alpha-alpha-hairpins with short connections. *Protein Eng.*, 4(3):245–250.

Eisenberg, D. & McLachlan, A. D. (**1986**). Solvation energy in protein folding and binding. *Nature*, 319(6050):199–203.

Elcock, A. H. (**2002**). Modeling supramolecular assemblages. *Curr.Opin.Struct.Biol.*, 12(2):154–160.

Elofsson, A. (**2002**). A study on protein sequence alignment quality. *Proteins*, 46(3):330–339.

Eyrich, V. A., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Fiser, A., Pazos, F., Valencia, A., Sali, A. & Rost, B. (**2001**). EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics*, 17(12):1242–1243.

Farooq, A., Chaturvedi, G., Mujtaba, S., Plotnikova, O., Zeng, L., Dhalluin, C., Ashton, R. & Zhou, M. M. (**2001**). Solution structure of ERK2 binding domain of MAPK phosphatase MKP-3: structural insights into MKP-3 activation by ERK2. *Mol.Cell.*, 7(2):387–399.

Fauman, E. B., Cogswell, J. P., Lovejoy, B., Rocque, W. J., Holmes, W., Montana, V. G., Piwnica-Worms, H., Rink, M. J. & Saper, M. A. (**1998a**). Crystal structure of the catalytic domain of the human cell cycle control phosphatase, Cdc25A. *Cell*, 93(4):617–625.

Fauman, E. B., Cogswell, J. P., Lovejoy, B., Rocque, W. J., Holmes, W., Montana, V. G., Piwnica-Worms, H., Rink, M. J. & Saper, M. A. (**1998b**). Crystal structure of the catalytic domain of the human cell cycle control phosphatase, Cdc25A. *Cell*, 93(4):617–625.

Fedorov, A., Cao, X., Saxonov, S., de Souza., S. J., Roy, S. W. & Gilbert, W. (**2001**). Intron distribution difference for 276 ancient and 131 modern genes suggests the existence of ancient introns. *PNAS*, 98(23):13177–13182.

Fedorov, A., Merican, A. F. & Gilbert, W. (**2002**). Large-scale comparison of intron positions among animal, plant, and fungal genes. *PNAS*, 99(25):16128–16133.

Feng, D. F. & Doolittle, R. F. (**1987**). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J.Mol.Evol.*, 25(4):351–360.

Fiaux, J., Bertelsen, E. B., Horwich, A. L. & Wuthrich, K. (**2002**). NMR analysis of a 900K GroEL GroES complex. *Nature*, 418(6894):207–211.

Fidelis, K., Stern, P. S., Bacon, D. & Moult, J. (**1994**). Comparison of systematic search and database methods for constructing segments of protein structure. *Protein Eng.*, 7(8):953–960.

Fischer, D. (**2000**). Hybrid fold recognition: combining sequence derived properties with evolutionary information. *Pac.Symp.Biocomput.*, pages 119–130.

Fischer, D. (**2003**). 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins*, 51(3):434–441.

Fiser, A., Do, R. K. & Sali, A. (**2000**). Modeling of loops in protein structures. *Protein Sci.*, 9(9):1753–1773.

Frishman, D. & Argos, P. (**1995**). Knowledge-based protein secondary structure assignment. *Proteins*, 23(4):566–579.

Fukushima, K., Kikuchi, J., Koshiba, S., Kigawa, T., Kuroda, Y. & Yokoyama, S. (**2002**). Solution structure of the DFF-C domain of DFF45/ICAD. A structural basis for the regulation of apoptotic DNA fragmentation. *J.Mol.Biol.*, 321(2):317–327.

Garmendia, J., Devos, D., Valencia, A. & de Lorenzo, V. (**2001**). A la carte transcriptional regulators: unlocking responses of the prokaryotic enhancer-binding protein XylR to non-natural effectors. *Mol.Microbiol.*, 42(1):47–59.

Gershenfeld, N. (**1999**). *The Nature of Mathematical Modelling*. Cambridge University Press.

Gerstein, M. & Levitt, M. (**1996**). Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. *Proc.Int.Conf.Intell.Syst.Mol.Biol.*, 4:59–67.

Gilbert, W. (**1978**). Why genes in pieces? *Nature*, 271:501.

Ginalski, K., Elofsson, A., Fischer, D. & Rychlewski, L. (**2003**). 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, 19(8):1015–1018.

Gong, H., Isom, D. G., Srinivasan, R. & Rose, G. D. (**2003**). Local secondary structure content predicts folding rates for simple, two-state proteins. *J.Mol.Biol.*, 327(5):1149–1154.

Gonnet, G., Cohen, M. & Benner, S. (**1992**). Exhaustive matching of the entire protein sequence database. *Science*, 256(5062):1443–1445.

Gonzalo-Arroyo, J. & Rodríguez-Artacho, M. (**1997**). *Esquemas algorítmicos: enfoque metodológico y problemas resueltos.*. Universidad Nacional de Educación a Distancia, Madrid, 1st edition.

Gotoh, O. (**1982**). An improved algorithm for matching biological sequences. *J.Mol.Biol.*, 162(3):705–708.

Greer, J. (**1981**). Comparative model-building of the mammalian serine proteases. *J.Mol.Biol.*, 153(4):1027–1042.

Guex, N., Diemand, A. & Peitsch, M. C. (**1999**). Protein modelling for all. *Trends Biochem.Sci.*, 24(9):364–367.

Halgren, T. A. & Damm, W. (**2001**). Polarizable force fields. *Curr.Opin.Struct.Biol.*, 11(2):236–242.

Hall, P. R., Wang, Y.-F., Rivera-Hainaj, R. E., Zheng, X., Pustai-Carey, M., Carey, P. R. & Yee, V. C. (**2003**). Transcarboxylase 12S crystal structure: hexamer assembly and substrate binding to a multienzyme core. *EMBO J.*, 22(10):2334–2347.

Hartl, F. U. & Hayer-Hartl, M. (**2002**). Molecular chaperones in the cytosol: from nascent chain to folded protein. *Science*, 295(5561):1852–1858.

Havel, T. F. & Snow, M. E. (**1991**). A new method for building protein conformations from sequence alignments with homologues of known structure. *J.Mol.Biol.*, 217(1):1–7.

Henderson, R., Baldwin, J. M., Ceska, T. A., Zemlin, F., Beckmann, E. & Downing, K. H. (**1990**). Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *J.Mol.Biol.*, 213(4):899–929.

Henikoff, S. & Henikoff, J. G. (**1992**). Amino acid substitution matrices from protein blocks. *PNAS*, 89(22):10915–10919.

Henikoff, S. & Henikoff, J. G. (**1993**). Performance evaluation of amino acid substitution matrices. *Proteins*, 17(1):49–61.

Henikoff, S., Henikoff, J. G. & Pietrokovski, S. (**1999**). Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, 15(6):471–479.

Hogg, N. & Bates, P. A. (**2000**). Genetic analysis of integrin function in man: LAD-1 and other syndromes. *Matrix Biol.*, 19(3):211–222.

Holland, J. (**1975**). *Adaptation in Natural and Artificial Systems.*. University of Michigan Press., Ann Arbor.

Holm, L. & Sander, C. (**1992**). Evaluation of protein models by atomic solvation preference. *J.Mol.Biol.*, 225(1):193–105.

Hooft, R., Sander, C. & Vriend, G. (**1996**). Verification of protein structures: side-chain planarity. *J.Appl.Cryst.*, 29:714–716.

Hu, J., Shen, X., Shao, Y., Bystroff, C. & Zaki, M. J. (**2002**). Mining Protein Contact Maps. *BIOKDD02: Workshop on Data Mining in Bioinformatics*, pages 3–10.

Hubbard, S. & Thornton, J. M. (**1993**). NACCESS, Computer program. *Department of Biochemistry and Molecular Biology, University College, London.*.

Hutchinson, E. G. & Thornton, J. M. (**1994**). A revised set of potentials for beta-turn formation in proteins. *Protein Sci.*, 3(12):2207–2216.

Huyton, T., Bates, P. A., Zhang, X., Sternberg, M. J. & Freemont, P. S. (**2000**). The BRCA1 C-terminal domain: structure and function. *Mutat.Res.*, 460(3-4):319–332.

Jacobson, M. P., Friesner, R. A., Xiang, Z. & Honig, B. (**2002**). On the role of the crystal environment in determining protein side- chain conformations. *J.Mol.Biol.*, 320(3):597–608.

Janardhan, A. & Vajda, S. (**1998**). Selecting near-native conformations in homology modeling: the role of molecular mechanics and solvation terms. *Protein Sci.*, 7(8):1772–1780.

Janin, J., Henrick, K., Moult, J., Eyck, L. T., Sternberg, M. J. E., Vajda, S., Vakser, I. & Wodak, S. J. (**2003**). CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins*, 52(1):2–9.

John, B. & Sali, A. (**2003**). Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic Acids Res.*, 31(14):3982–3992.

Johnson, W. C. J. (**1990**). Protein secondary structure and circular dichroism: a practical guide. *Proteins*, 7(3):205–214.

Jones, D. T. (**1999**). Protein secondary structure prediction based on position-specific scoring matrices. *J.Mol.Biol.*, 292(2):195–202.

Jones, D. T. (**2001**). Predicting novel protein folds by using FRAGFOLD. *Proteins*, Suppl 5:127–132.

Jones, D. T., Taylor, W. R. & Thornton, J. M. (**1992**). A new approach to protein fold recognition. *Nature*, 358(6381):86–89.

Jones, T. A. & Thirup, S. (**1986**). Using known substructures in protein model building and crystallography. *EMBO J.*, 5(4):819–822.

Kabsch, W. & Sander, C. (**1983**). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(2577):2577–2637.

Karlin, S. & Altschul, S. F. (**1990**). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *PNAS*, 87(6):2264–2268.

Karlin, S. & Altschul, S. F. (**1993**). Applications and statistics for multiple high-scoring segments in molecular sequences. *PNAS*, 90(12):5873–5877.

Katona, G., Berglund, G. I., Hajdu, J., Graf, L. & Szilagyi, L. (**2002**). Crystal structure reveals basis for the inhibitor resistance of human brain trypsin. *J.Mol.Biol.*, 315(5):1209–1218.

Kawasaki, H. & Kretsinger, R. H. (**1995**). Calcium-binding proteins 1: EF-hands. *Protein Profile*, 2(4):297–490.

Keasar, C. & Levitt, M. (**2003**). A novel approach to decoy set generation: designing a physical energy function having local minima with native structure characteristics. *J.Mol.Biol.*, 329(1):159–174.

Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (**2000**). Enhanced genome annotation using structural profiles in the program 3D- PSSM. *J.Mol.Biol.*, 299(2):499–520.

Kent, W. J. (**2002**). BLAT–the BLAST-like alignment tool. *Genome Res.*, 12(4):656–664.

Koehl, P. & Delarue, M. (**1994**). Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J.Mol.Biol.*, 239(2):249–275.

Koehl, P. & Levitt, M. (**1999**). De novo protein design. I. In search of stability and specificity. *J.Mol.Biol.*, 293(5):1161–1181.

Koehl, P. & Levitt, M. (**2002**). Sequence variations within protein families are linearly related to structural variations. *J.Mol.Biol.*, 323:551–562.

Kolatkar, P. R., Bella, J., Olson, N. H., Bator, C. M., Baker, T. S. & Rossmann, M. G. (**1999**). Structural studies of two rhinovirus serotypes complexed with fragments of their cellular receptor. *EMBO J.*, 18(22):6249–6259.

Kolodny, R., Koehl, P., Guibas, L. & Levitt, M. (**2002**). Small Libraries of Protein Fragments Model Native Protein Structures Accurately. *J.Mol.Biol.*, 323:297–307.

Kraulis, P. (**1991**). MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. *J. Appl. Cryst.*, 24:946–950.

Krogh, A., Brown, M., Mian, I. S., Sjolander, K. & Haussler, D. (**1994**). Hidden Markov models in computational biology. Applications to protein modeling. *J.Mol.Biol.*, 235(5):1501–1531.

Kussell, E., Shimada, J. & Shakhnovich, E. I. (**2001**). Excluded volume in protein side-chain packing. *J.Mol.Biol.*, 311(1):183–193.

Kwasigroch, J. M., Chomilier, J. & Mornon, J. P. (**1996**). A global taxonomy of loops in globular proteins. *J.Mol.Biol.*, 259(4):855–872.

Lah, M. S., Dixon, M. M., Pattridge, K. A., Stallings, W. C., Fee, J. A. & Ludwig, M. L. (**1995**). Structure-function in Escherichia coli iron superoxide dismutase: comparisons with the manganese enzyme from Thermus thermophilus. *Biochemistry*, 34(5):1646–1660.

Lambert, C., Leonard, N., De Bolle, X. & Depiereux, E. (**2002**). ESyPred3D: Prediction of proteins 3D structures. *Bioinformatics*, 18(9):1250–1256.

Laskowski, R. A., MacArthur, M. W., Moss, D. & Thornton, J. M. (**1993**). PROCHECK: a program to check the stereochemical quality of protein structures. *J.Appl.Cryst.*, 26:283–291.

Lasters, I. & Desmet, J. (**1993**). The fuzzy-end elimination theorem: correctly implementing the side chain placement algorithm based on the dead-end elimination theorem. *Protein Eng.*, 6(7):717–722.

Leach, A. R. (**2001**). *Molecular modelling: principles and applications.*. Prentice Hall, 2nd edition.

Lee, C. & Subbiah, S. (**1991**). Prediction of protein side-chain conformation by packing optimization. *J.Mol.Biol.*, 217(2):373–388.

Lee, M. R., Tsai, J., Baker, D. & Kollman, P. A. (**2001**). Molecular dynamics in the endgame of protein structure prediction. *J.Mol.Biol.*, 313(2):417–430.

Lehninger, A. L. (**1982**). *Principles of Biochemistry*. Worth Pub., New York, 1st edition.

Levitt, M. (**1976**). A simplified representation of protein conformations for rapid simulation of protein folding. *J.Mol.Biol.*, 104(1):59–107.

Liang, S. & Grishin, N. V. (**2002**). Side-chain modeling with an optimized scoring function. *Protein Sci.*, 11(2):322–331.

Liepinsh, E., Genereux, C., Dehareng, D., Joris, B. & Otting, G. (**2003**). NMR structure of Citrobacter freundii AmpD, comparison with bacteriophage T7 lysozyme and homology with PGRP domains. *J.Mol.Biol.*, 327(4):833–842.

Lindahl, E., Hess, B. & van der Spoel, D. (**2001**). GROMACS 3.0: a package for molecular simulation and trajectory analysis. *J.Mol.Model.*, 7:306–317.

Liu, J., Tan, H. & Rost, B. (**2002**). Loopy proteins appear conserved in evolution. *J.Mol.Biol.*, 322(1):53–64.

Looger, L. L., Dwyer, M. A., Smith, J. J. & Hellinga, H. W. (**2003**). Computational design of receptor and sensor proteins with novel functions. *Nature*, 423(6936):185–190.

Looger, L. L. & Hellinga, H. W. (**2001**). Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *J.Mol.Biol.*, 307(1):429–445.

Lund, O., Frimand, K., Gorodkin, J., Bohr, H., Bohr, J., Hansen, J. & Brunak, S. (**1997**). Protein distance constraints predicted by neural networks and probability density functions. *Protein Eng.*, 10(11):1241–1248.

Lundstrom, J., Rychlewski, L., Bujnicki, J. & Elofsson, A. (**2001**). Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci.*, 10(11):2354–2362.

Luz, J. G., Hassig, C. A., Pickle, C., Godzik, A., Meyer, B. J. & Wilson, I. A. (**2003**). XOL-1, primary determinant of sexual fate in C. elegans, is a GHMP kinase family member and a structural prototype for a class of developmental regulators. *Genes Dev.*, 17(8):977–990.

Mangoni, M., Roccatano, D. & Di Nola, A. (**1999**). Docking of flexible ligands to flexible receptors in solution by molecular dynamics simulation. *Proteins*, 35(2):153–162.

Mansfeld, J., Vriend, G., Dijkstra, B. W., Veltman, O. R., Van den Burg, B., Venema, G., Ulbrich-Hofmann, R. & Eijsink, V. G. (**1997**). Extreme stabilization of a thermolysin-like protease by an engineered disulfide bond. *J.Biol.Chem.*, 272(17):11152–11156.

Marti-Renom, M. A., Madhusudhan, M. S., Fiser, A., Rost, B. & Sali, A. (**2002**). Reliability of assessment of protein structure prediction methods. *Structure*, 10(3):435–440.

Marti-Renom, M. A., Stuart, A. C., Fiser, A., Sanchez, R., Melo, F. & Sali, A. (**2000**). Comparative protein structure modeling of genes and genomes. *Annu.Rev.Biophys.Biomol.Struct.*, 29:291–325.

Martin, A. C., MacArthur, M. W. & Thornton, J. M. (**1997**). Assessment of comparative modeling in CASP2. *Proteins*, Suppl(1):14–28.

May, A. C. & Johnson, M. S. (**1994**). Protein structure comparisons using a combination of a genetic algorithm, dynamic programming and least-squares minimization. *Protein Eng.*, 7(4):475–485.

May, A. C. & Johnson, M. S. (**1995**). Improved genetic algorithm-based protein structure comparisons: pairwise and multiple superpositions. *Protein Eng.*, 8(9):873–882.

McGregor, M. J., Islam, S. A. & Sternberg, M. J. (**1987**). Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *J.Mol.Biol.*, 198(2):295–310.

McLachlan, A. D. (**1979**). Gene duplications in the structural evolution of chymotrypsin. *J.Mol.Biol.*, 128(1):49–79.

Melo, F., Sanchez, R. & Sali, A. (**2002**). Statistical potentials for fold assessment. *Protein Sci.*, 11(2):430–448.

Mendes, J., Baptista, A. M., Carrondo, M. A. & Soares, C. M. (**1999**). Improved modeling of side-chains in proteins with rotamer-based methods: a flexible rotamer model. *Proteins*, 37(4):530–543.

Mewes, H. (**1991**). MIPS - European node for protein sequence data. *CODATA Bulletin*, 23:62–63.

Mezei, M. (**1998**). Chameleon sequences in the PDB. *Protein Eng.*, 11(6):411–414.

Michalewicz, Z. (**1996**). *Genetic algorithms + data structures = evolution programs.* Springer-Verlag, New York, 3rd rev. and extended edition.

Miller, D. J., Ouellette, N., Evdokimova, E., Savchenko, A., Edwards, A. & Anderson, W. F. (**2003**). Crystal complexes of a predicted S-adenosylmethionine-dependent methyltransferase reveal a typical AdoMet binding domain and a substrate recognition domain. *Protein Sci.*, 12(7):1432–1442.

Milner-White, E. J. & Poet, R. (**1986**). Four classes of beta-hairpins in proteins. *Biochem J.*, 240(1):289–292.

Montelione, G. T., Zheng, D., Huang, Y. J., Gunsalus, K. C. & Szyperski, T. (**2000**). Protein NMR spectroscopy in structural genomics. *Nat.Struct.Biol.*, 7 Suppl:982–985.

Morris, G. M., Goodsell, D. S., Huey, R. & Olson, A. J. (**1996**). Distributed automated docking of flexible ligands to proteins: parallel applications of AutoDock 2.4. *J.Comput.Aided Mol.Des.*, 10(4):293–304.

Mourier, T. & Jeffares, D. C. (**2003**). Eukaryotic intron loss. *Science*, 300(5624):1393.

Muller, A., MacCallum, R. M. & Sternberg, M. J. E. (**2002**). Structural characterization of the human proteome. *Genome Res.*, 12(11):1625–1641.

Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (**1995**). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J.Mol.Biol.*, 247(4):536–540.

Needleman, S. B. & Wunsch, C. D. (**1970**). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J.Mol.Biol.*, 48:443–453.

Notredame, C., Higgins, D. G. & Heringa, J. (**2000**). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J.Mol.Biol.*, 302(1):205–217.

Ogata, K. & Umeyama, H. (**2000**). An automatic homology modeling method consisting of database searches and simulated annealing. *J.Mol.Graph.Model.*, 18(3):258–272, 305–306.

Oliva, B., Bates, P. A., Querol, E., Aviles, F. X. & Sternberg, M. J. (**1997**). An automated classification of the structure of protein loops. *J.Mol.Biol.*, 266(4):814–830.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. & Thornton, J. M. (**1997**). CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108.

Padgett, R. A., Grabowski, P. J., Konarska, M. M., Seiler, S. & Sharp, P. A. (**1986**). Splicing of messenger RNA precursors. *Annu.Rev.Biochem.*, 55:1119–1150.

Padyana, A. K. & Burley, S. K. (**2003**). Crystal structure of shikimate 5-dehydrogenase (SDH) bound to NADP: insights into function and evolution. *Structure*, 11(8):1005–1013.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. & Chothia, C. (**1998**). Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J.Mol.Biol.*, 284(4):1201–1210.

Patthy, L. (**1999**). *Protein Evolution*. Blackwell Science, Oxford.

Pearson, W. R. & Lipman, D. J. (**1988**). Improved tools for biological sequence comparison. *PNAS*, 85(8):2444–2448.

Pedersen, J. T. & Moult, J. (**1995**). Ab initio structure prediction for small polypeptides and protein fragments using genetic algorithms. *Proteins*, 23(3):454–460.

Peitsch, M. C. (**2002**). About the use of protein models. *Bioinformatics*, 18(7):934–938.

Perona, J. J. & Craik, C. S. (**1997**). Evolutionary divergence of substrate specificity within the chymotrypsin-like serine protease fold. *J.Biol.Chem.*, 272(48):29987–29990.

Petersen, K. & Taylor, W. R. (**2003**). Modelling zinc-binding proteins with GADGET: genetic algorithm and distance geometry for exploring topology. *J.Mol.Biol.*, 325(5):1039–1059.

Petrella, R. J. & Karplus, M. (**2001**). The energetics of off-rotamer protein side-chain conformations. *J.Mol.Biol.*, 312(5):1161–1175.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (**1992**). *Numerical Recipes in C. The Art of Scientific Computing.*. Cambridge University Press, New York, 2nd edition.

Rabow, A. A. & Scheraga, H. A. (**1996**). Improved genetic algorithm for the protein folding problem by use of a Cartesian combination operator. *Protein Sci.*, 5(9):1800–1815.

Ramachandran, G. N. & Sasisekharan, V. (**1968**). Conformation of polypeptides and proteins. *Adv.Protein Chem.*, 23:283–438.

Reina, J., Lacroix, E., Hobson, S. D., Fernandez-Ballester, G., Rybin, V., Schwab, M. S., Serrano, L. & Gonzalez, C. (**2002**). Computer-aided design of a PDZ domain to recognize new target sequences. *Nat.Struct.Biol.*, 9(8):621–627.

Rhodes, G. (**2000**). *Crystallography Made Crystal Clear*. Academic Press, New York, 2nd edition.

Rice, P. A., Goldman, A. & Steitz, T. A. (**1990**). A helix-turn-strand structural motif common in alpha-beta proteins. *Proteins*, 8(4):334–340.

Riechmann, L. & Winter, G. (**2000**). Novel folded protein domains generated by combinatorial shuffling of polypeptide segments. *PNAS*, 97(18):10068–10073.

Robson, B. & Osguthorpe, D. J. (**1979**). Refined models for computer simulation of protein folding. Applications to the study of conserved secondary structure and flexible hinge points during the folding of pancreatic trypsin inhibitor. *J.Mol.Biol.*, 132(1):19–51.

Rose, G. D., Gierasch, L. M. & Smith, J. A. (**1985**). Turns in peptides and proteins. *Adv.Protein Chem.*, 37:1–109.

Rost, B. (**1995**). TOPITS: threading one-dimensional predictions into three-dimensional structures. *Proc.Int.Conf.Intell.Syst.Mol.Biol.*, 3:314–321.

Rost, B. (**1996**). PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol.*, 266:525–539.

Rost, B. & Eyrich, V. A. (**2001**). EVA: large-scale analysis of secondary structure prediction. *Proteins*, Suppl 5:192–199.

Russ, W. P. & Ranganathan, R. (**2002**). Knowledge-based potential functions in protein design. *Curr.Opin.Struct.Biol.*, 12(4):447–452.

Russell, R. B. & Barton, G. J. (**1992**). Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, 14(2):309–323.

Rychlewski, L., Jaroszewski, L., Li, W. & Godzik, A. (**2000**). Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci.*, 9(2):232–241.

Sali, A. & Blundell, T. L. (**1993**). Comparative protein modelling by satisfaction of spatial restraints. *J.Mol.Biol.*, 234(3):779–815.

Sali, A., Glaeser, R., Earnest, T. & Baumeister, W. (**2003**). From words to literature in structural proteomics. *Nature*, 422(6928):216–225.

Samudrala, R. & Moult, J. (**1998**). A graph-theoretic algorithm for comparative modeling of protein structure. *J.Mol.Biol.*, 279(1):287–302.

Sander, C. & Schneider, R. (**1991**). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9(1):56–68.

Sanger, F. (**1952**). The arrangement of amino acids in proteins. *Adv.Protein Chem.*, 7:1–67.

Sanishvili, R., Yakunin, A. F., Laskowski, R. A., Skarina, T., Evdokimova, E., Doherty-Kirby, A., Lajoie, G. A., Thornton, J. M., Arrowsmith, C. H., Savchenko, A., Joachimiak, A. & Edwards, A. M. (**2003**). Integrating structure, bioinformatics, and enzymology to discover function: BioH, a new carboxylesterase from Escherichia coli. *J.Biol.Chem.*, 278(28):26039–26045.

Saqi, M. A., Bates, P. A. & Sternberg, M. J. (**1992**). Towards an automatic method of predicting protein structure by homology: an evaluation of suboptimal sequence alignments. *Protein Eng.*, 5(4):305–311.

Saqi, M. A. & Sternberg, M. J. (**1991**). A simple method to generate non-trivial alternate alignments of protein sequences. *J.Mol.Biol.*, 219(4):727–732.

Sayle, R. & Milner-White, E. (**1995**). RASMOL: biomolecular graphics for all. *TIBS*, 20:374–376.

Schaffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V. & Altschul, S. F. (**2001**). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, 29(14):2994–3005.

Schaffer, A. A., Wolf, Y. I., Ponting, C. P., Koonin, E. V., Aravind, L. & Altschul, S. F. (**1999**). IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics*, 15(12):1000–1011.

Schafferhans, A. & Klebe, G. (**2001**). Docking ligands onto binding site representations derived from proteins built by homology modelling. *J.Mol.Biol.*, 307(1):407–427.

Schonbrun, J., Wedemeyer, W. J. & Baker, D. (**2002**). Protein structure prediction in 2002. *Curr.Opin.Struct.Biol.*, 12(3):348–354.

Schwartz, R. & Dayhoff, M. (**1978**). Matrices for detecting distant relationships. *Atlas of Protein Sequence and Structure*, 5 Suppl.(3):353–358. Natl. Biomed. Res. Found., Washington, DC.

Sedgewick, R. (**1988**). *Algorithms*. Addison-Wesley, 2nd edition.

Sellar, G. C., Watt, K. P., Rabiasz, G. J., Stronach, E. A., Li, L., Miller, E. P., Massie, C. E., Miller, J., Contreras-Moreira, B., Scott, D., Brown, I., Williams, A. R., Bates, P. A., Smyth, J. F. & Gabra, H. (**2003**). OPCML at 11q25 is epigenetically inactivated and has tumor-suppressor function in epithelial ovarian cancer. *Nat.Genet.*, 34(3):337–343.

Sellers, P. H. (**1974**). On the theory and computation of evolutionary distances. *SIAM J.Appl.Math.*, 26:787–793.

Shaanan, B., Gronenborn, A. M., Cohen, G. H., Gilliland, G. L., Veerapandian, B., Davies, D. R. & Clore, G. M. (**1992**). Combining experimental information from crystal and solution studies: joint X-ray and NMR refinement. *Science*, 257(5072):961–964.

Shi, J., Blundell, T. L. & Mizuguchi, K. (**2001**). FUGUE: sequence-structure homology recognition using environment- specific substitution tables and structure-dependent gap penalties. *J.Mol.Biol.*, 310(1):243–257.

Shindyalov, I. N. & Bourne, P. E. (**1998**). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, 11(9):739–747.

Simons, K. T., Bonneau, R., Ruczinski, I. & Baker, D. (**1999**). Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins*, Suppl(3):171–176.

Simons, K. T., Kooperberg, C., Huang, E. & Baker, D. (**1997a**). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J.Mol.Biol.*, 268(1):209–225.

Simons, K. T., Kooperberg, C., Huang, E. & Baker, D. (**1997b**). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J.Mol.Biol.*, 268(1):209–225.

Sippl, M. J. (**1990**). Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *J.Mol.Biol.*, 213(4):859–883.

Sippl, M. J., Lackner, P., Domingues, F. S., Prlic, A., Malik, R., Andreeva, A. & Wiederstein, M. (**2001**). Assessment of the CASP4 fold recognition category. *Proteins*, Suppl(5):55–67.

Smith, T. & Waterman, M. (**1981**). Identification of Common Molecular Subsequences. *J.Mol.Biol.*, 147:195–197.

Sonnhammer, E. L., Eddy, S. R., Birney, E., Bateman, A. & Durbin, R. (**1998**). Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res.*, 26(1):320–322.

Stewart, A. E., Dowd, S., Keyse, S. M. & McDonald, N. Q. (**1999**). Crystal structure of the MAPK phosphatase Pystl catalytic domain and implications for regulated activation. *Nat.Struct.Biol.*, 6(2):174–181.

Stoltzfus, A., Logsdon, J. M. J., Palmer, J. D. & Doolittle, W. F. (**1997**). Intron "sliding" and the diversity of intron positions. *PNAS*, 94(20):10739–10744.

Stoltzfus, A., Spencer, D. F., Zuker, M., Logsdon, J. M. J. & Doolittle, W. F. (**1994**). Testing the exon theory of genes: the evidence from protein structure. *Science*, 265(5169):202–207.

Suguna, K., Bott, R. R., Padlan, E. A., Subramanian, E., Sheriff, S., Cohen, G. H. & Davies, D. R. (**1987**). Structure and refinement at 1.8 A resolution of the aspartic proteinase from Rhizopus chinensis. *J.Mol.Biol.*, 196(4):877–900.

Sutcliffe, M. J., Haneef, I., Carney, D. & Blundell, T. L. (**1987a**). Knowledge based modelling of homologous proteins, Part I: Three- dimensional frameworks derived from the simultaneous superposition of multiple structures. *Protein Eng.*, 1(5):377–384.

Sutcliffe, M. J., Hayes, F. R. & Blundell, T. L. (**1987b**). Knowledge based modelling of homologous proteins, Part II: Rules for the conformations of substituted sidechains. *Protein Eng.*, 1(5):385–392.

Taylor, P., Bilsland, M. & Walkinshaw, M. D. (**2001**). A new conformation of the integrin-binding fragment of human VCAM-1 crystallizes in a highly hydrated packing arrangement. *Acta Crystallogr. D Biol.Crystallogr.*, 57(Pt 11):1579–1583.

Taylor, W. R. (**2001**). Defining linear segments in protein structure. *J.Mol.Biol.*, 310(5):1135–1150.

Taylor, W. R. & Orengo, C. A. (**1989**). Protein structure alignment. *J.Mol.Biol.*, 208(1):1–22.

Teplyakov, A., Obmolova, G., Khil, P. P., Howard, A. J., Camerini-Otero, R. D. & Gilliland, G. L. (**2003**). Crystal structure of the Escherichia coli YcdX protein reveals a trinuclear zinc active site. *Proteins*, 51(2):315–318.

Thompson, J. D., Higgins, D. G. & Gibson, T. J. (**1994**). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22(22):4673–4680.

Tovchigrechko, A., Wells, C. A. & Vakser, I. A. (**2002**). Docking of protein models. *Protein Sci.*, 11(8):1888–1896.

Tramontano, A., Leplae, R. & Morea, V. (**2001**). Analysis and assessment of comparative modeling predictions in CASP4. *Proteins*, Suppl(5):22–38.

Tress, M., Jones, D. & Valencia, A. (**2003**). Predicting Reliable Regions in Protein Alignments from Sequence Profiles. *J.Mol.Biol.*, 330(4):705–718.

Unger, R., Harel, D., Wherland, S. & Sussman, J. L. (**1989**). A 3D building blocks approach to analyzing and predicting structure of proteins. *Proteins*, 5(4):355–373.

Unger, R. & Moult, J. (**1993**). Genetic algorithms for protein folding simulations. *J.Mol.Biol.*, 231(1):75–81.

van Vlijmen, H. W. & Karplus, M. (**1997**). PDB-based protein loop prediction: parameters for selection and methods for optimization. *J.Mol.Biol.*, 267(4):975–1001.

Venclovas, C. (**2001**). Comparative modeling of CASP4 target proteins: combining results of sequence search with three-dimensional structure assessment. *Proteins*, Suppl(5):47–54.

Ventkatachalam, C. M. (**1968**). Stereochemical criteria for polypeptides and proteins. Conformation of a system of three linked peptide units. *Biopolymers*, 6:1425–1436.

Vitkup, D., Melamud, E., Moult, J. & Sander, C. (**2001**). Completeness in structural genomics. *Nat.Struct.Biol.*, 8(6):559–566.

Voigt, C. A., Martinez, C., Wang, Z.-G., Mayo, S. L. & Arnold, F. H. (**2002**). Protein building blocks preserved by recombination. *Nat.Struct.Biol.*, 9(7):553–558.

Wagner, G. & Wuthrich, K. (**1982**). Sequential resonance assignments in protein 1H nuclear magnetic resonance spectra. Basic pancreatic trypsin inhibitor. *J.Mol.Biol.*, 155(3):347–366.

Walker, J. E., Saraste, M., Runswick, M. J. & Gay, N. J. (**1982**). Distantly related sequences in the alpha- and beta-subunits of ATP synthase, myosin, kinases and other ATP-requiring enzymes and a common nucleotide binding fold. *EMBO J.*, 1(8):945–951.

Wallner, B. & Elofsson, A. (**2003**). Can correct protein models be identified. *Protein Sci.*, 12(5):1073–1086.

Wang, S. & Eisenberg, D. (**2003**). Crystal structures of a pantothenate synthetase from M. tuberculosis and its complexes with substrates and a reaction intermediate. *Protein Sci.*, 12(5):1097–1108.

Wang, Z. & Moult, J. (**2001**). SNPs, protein structure, and disease. *Hum.Mutat.*, 17(4):263–270.

Waterman, M. S. (**1995**). *Introduction to computational biology*. Chapman and Hall, New York, NY. 94036997 Michael S. Waterman.

Whitson, R. H., Huang, T. & Itakura, K. (**1999**). The novel Mrf-2 DNA-binding domain recognizes a five-base core sequence through major and minor-groove contacts. *Biochem.Biophys.Res.Commun.*, 258(2):326–331.

Williams, S. B., Vakonakis, I., Golden, S. S. & LiWang, A. C. (**2002**). Structure and function from the circadian clock protein KaiA of Synechococcus elongatus: a potential clock input mechanism. *PNAS*, 99(24):15357–15362.

Wintjens, R. T., Rooman, M. J. & Wodak, S. J. (**1996**). Automatic classification and analysis of alpha alpha-turn motifs in proteins. *J.Mol.Biol.*, 255(1):235–253.

Wood, T. C. & Pearson, W. R. (**1999**). Evolution of protein sequences and structures. *J.Mol.Biol.*, 291(4):977–995.

Wootton, J. C. & Federhen, S. (**1996**). Analysis of compositionally biased regions in sequence databases. *Methods Enzymol.*, 266:554–571.

Wriggers, W. & Chacon, P. (**2001**). Modeling tricks and fitting techniques for multiresolution structures. *Structure*, 9(9):779–788.

Wu, C. H., Yeh, L.-S. L., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z., Kourtesis, P., Ledley, R. S., Suzek, B. E., Vinayaka, C. R., Zhang, J. & Barker, W. C. (**2003**). The Protein Information Resource. *Nucleic Acids Res.*, 31(1):345–347.

Xia, Y. & Levitt, M. (**2002**). Roles of mutation and recombination in the evolution of protein thermodynamics. *PNAS*, 99(16):10382–10387.

Xiang, Z. & Honig, B. (**2001**). Extending the accuracy limits of prediction for side-chain conformations. *J.Mol.Biol.*, 311(2):421–430.

Xiang, Z., Soto, C. S. & Honig, B. (**2002**). Evaluating conformational free energies: the colony energy and its application to the problem of loop prediction. *PNAS*, 99(11):7432–7437.

Yamashita, A., Maeda, K. & Maeda, Y. (**2003**). Crystal structure of CapZ: structural basis for actin filament barbed end capping. *EMBO J.*, 22(7):1529–1538.

Yuan, Y. C., Whitson, R. H., Liu, Q., Itakura, K. & Chen, Y. (**1998**). A novel DNA-binding motif shares structural homology to DNA replication and repair nucleases and polymerases. *Nat.Struct.Biol.*, 5(11):959–964.

Zagrovic, B., Snow, C., Shirts, M. & Pande, V. (**2002**a). Simulation of folding of a small alpha-helical protein in atomistic detail using worldwide-distributed computing. *J.Mol.Biol.*, 323:927–937.

Zagrovic, B., Snow, C. D., Khaliq, S., Shirts, M. R. & Pande, V. S. (**2002**b). Native-like mean structure in the unfolded ensemble of small proteins. *J.Mol.Biol.*, 323(1):153–164.

Zemla, A. (**2003**). LGA: A method for finding 3D similarities in protein structures. *Nucl.Acids Res.*, 31(13):3370–3374.

Zhang, R.-g., Grembecka, J., Vinokour, E., Collart, F., Dementieva, I., Minor, W. & Joachimiak, A. (**2002**). Structure of Bacillus subtilis YXKO–a member of the UPF0031 family and a putative kinase. *J.Struct.Biol.*, 139(3):161–170.

Zhang, X., Shaw, A., Bates, P. A., Newman, R. H., Gowen, B., Orlova, E., Gorman, M. A., Kondo, H., Dokurno, P., Lally, J., Leonard, G., Meyer, H., van Heel, M. & Freemont, P. S. (**2000**). Structure of the AAA ATPase p97. *Mol.Cell.*, 6(6):1473–1484.

Zhu, L., Hu, J., Lin, D., Whitson, R., Itakura, K. & Chen, Y. (**2001**). Dynamics of the Mrf-2 DNA-binding domain free and in complex with DNA. *Biochemistry*, 40(31):9142–9150.